NUTS AND VOLTS
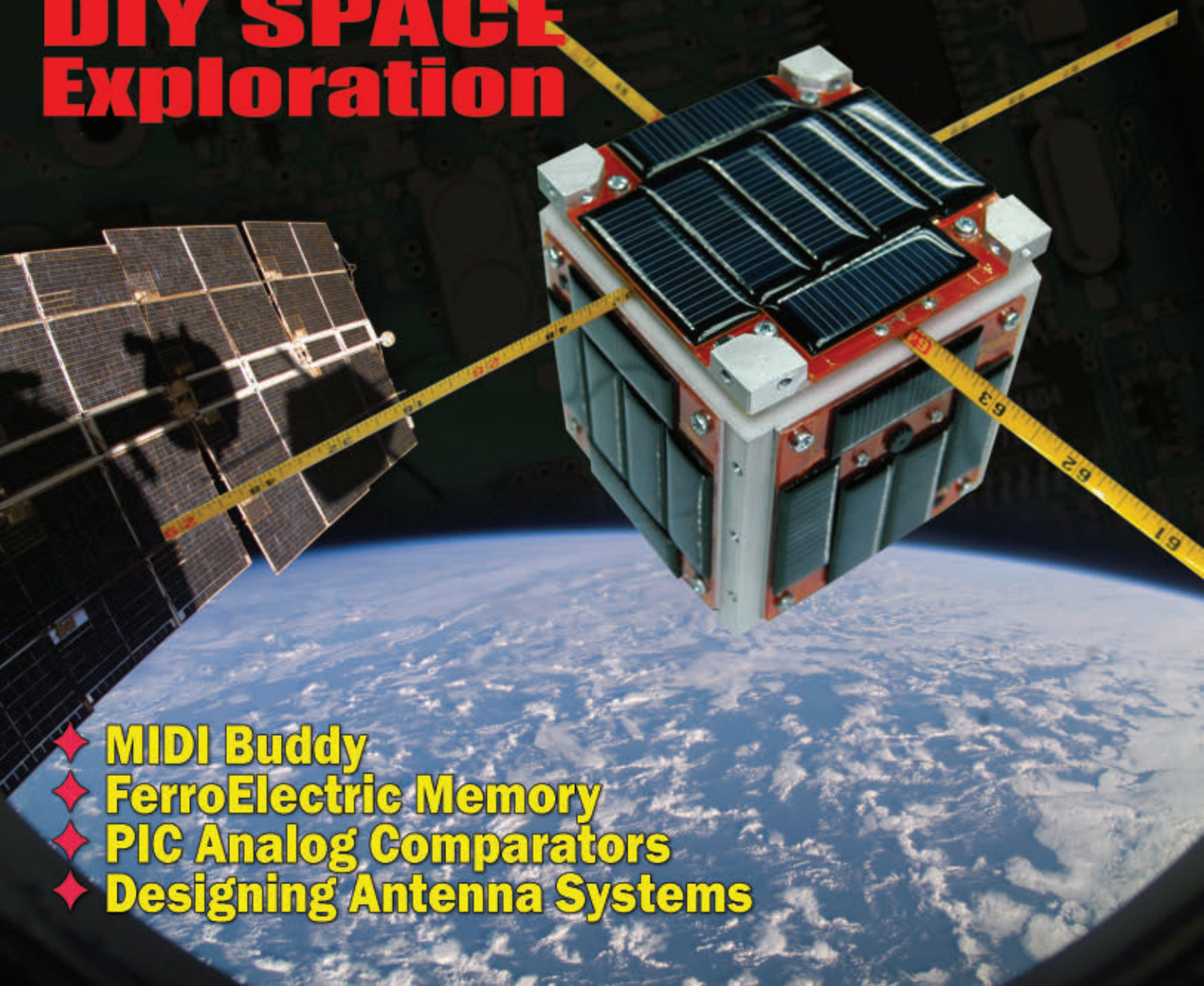
www.nutsvolts.com
February 2015

EVERYTHING FOR ELECTRONICS

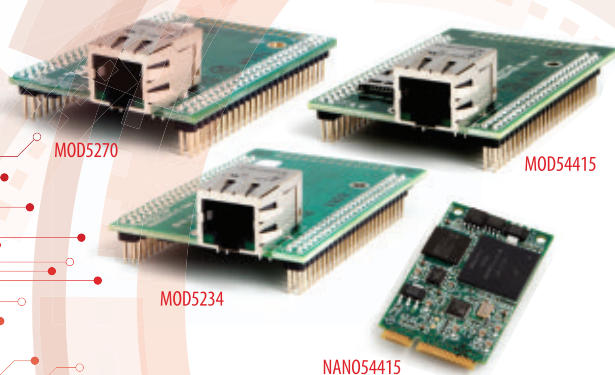## CubeSats
The latest in
DIY SPACE
Exploration

Practical 3D Printing
Real World Uses For The
Electronics Experimenter

◆ **MIDI Buddy**
◆ **FerroElectric Memory**
◆ **PIC Analog Comparators**
◆ **Designing Antenna Systems**

Ethernet Core Modules with
High-Performance Connectivity Options

MOD5270

MOD54415

MOD5234

NANO54415

➤ **MOD5270**
147.5 MHz processor with 512KB Flash & 8MB RAM · 47 GPIO
3 UARTs · I²C · SPI

➤ **MOD5234**
147.5 MHz processor with 2MB flash & 8MB RAM · 49 GPIO · 3 UARTs
I²C · SPI · CAN · eTPU (for I/O handling, serial communications,
motor/timing/engine control applications)

➤ **MOD54415**
250 MHz processor with 32MB flash & 64MB RAM · 42 GPIO · 8 UARTs
5 I²C · 3 SPI · 2 CAN · SSI · 8 ADC · 2 DAC · 8 PWM · 1-Wire® interface

➤ **NANO54415**
250 MHz processor with 8MB flash & 64MB RAM · 30 GPIO · 8 UARTs
4 I²C · 3 SPI · 2 CAN · SSI · 6 ADC · 2 DAC · 8 PWM · 1-Wire® interface

Add Ethernet connectivity to an existing product, or use it as your product's core processor
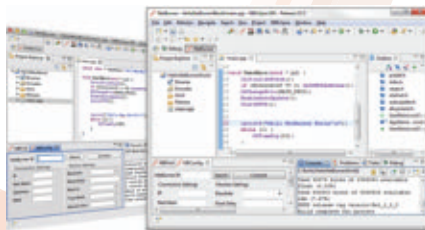
The goal: Control, configure, or monitor a device using Ethernet

The method: Create and deploy applications from your Mac or Windows PC. Get hands-on familiarity with the NetBurner platform by studying, building, and modifying source code examples.

The result: Access device from the Internet or a local area network (LAN)

The NetBurner Ethernet Core Module is a device containing everything needed for design engineers to add network control and to monitor a company's communications assets. For a very low price point, this module solves the problem of network-enabling devices with 10/100 Ethernet, including those requiring digital, analog and serial control.

MOD5270-100IR.........$69 (qty. 100)
MOD5234-100IR.........$99 (qty. 100)
MOD54415-100IR......$89 (qty. 100)
NANO54415-200IR...$69 (qty. 100)

NNDK-MOD5270LC-KIT..............$99
NNDK-MOD5234LC-KIT............$249
NNDK-MOD54415LC-KIT..........$129
NNDK-NANO54415-KIT..............$99

NetBurner Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications. The kits include all the hardware and software you need to build your embedded application.

➤ **For additional information please visit**
http://www.netburner.com/kits

NetBurner
Networking in One Day!

Information and Sales | sales@netburner.com
Web | www.netburner.com
Telephone | 1-800-695-6828

freescale™
Alliance Member

all the inventions waiting to happen...

CLICKER 2 STM32

price: $39⁰⁰

February 2015

Columns

Departments

One-Upmanship

An annual event in my old ham radio club was to hold a drawing for what began as a homebrew straight key gifted to a newcomer to the club. The original key — a momentary on switch — wasn't much more than a 3" copper bar, a thumb-tack contact, and a spring.

When the novice could afford a commercial key, he returned it to the original owner — with a few modifications. The improved key had a nice brass knob for the fingertips, and the flimsy base was replaced with a substantial slab of oak.

The key was passed on to the next needy club member in like fashion, with the expectation that it would be returned with some significant improvement. As I recall, by the fifth or sixth iteration of the gift/improvement process, the straight key had morphed into an iambic keyer with built-in sidetone oscillator.

That is, instead of pressing down on a fancy momentary open switch, the operator used the keyer by gently touching one pad with the thumb and the other pad with the index finger. Because finger motion was used instead of wrist and arm motion to operate the keyer, the result was much faster keying speeds.

Benefits of iambic keyers vs. straight keys aside, the point is that this little game of one-upmanship was one of the highpoints of the club. Everyone looked forward to their chance to demonstrate their prowess in circuit and mechanical design. It certainly was more entertaining — and challenging — than simply building a circuit according to a magazine article or duplicating a circuit developed by another club member.

In different issues of *Nuts & Volts*, you'll find articles that build on the work of others. In some ways, these can be considered a form of one-upmanship. In other ways, these articles are opportunities for you to demonstrate your ability to one-up previous writers. Sure, go ahead and build one of the projects. But don't stop there. See what you can do to improve on it — whether that involves making it simpler and more elegant, or adding a few new features.

Better yet, pass your handiwork off to a fellow experimenter and challenge them to one-up your work. In the end, everyone wins. **NV**

READER FEEDBACK

Earrings are a Hit

Thanks for the neat Christmas earring kit/article in the December 2014 issue by Ron Newton. It was my first time with SMD and it went very well.

I only lost one part: the .1 cap. It clicked in my tweezers and that was the last I saw of it.

Luckily, I found one on an old junk board. Thanks again! My wife loved them!

Dave Smith

Love of Tube Technology

I am bemused by the resurgence of interest in vacuum tube technology. I wonder how many tube enthusiasts had to design with them in their heyday.

As a teenager with a passion for electronics, I struggled to make amplifiers with greater output power and less distortion using 6L6s, premium EL34s, and even transmitting 807 tubes. I can still remember these numbers.

During a summer job at the English labs of Philips, I was one of the first in England to get my hands on transistors. I was eager to embrace this new technology.

Back in 1955, I made probably the very first pocket-sized transistor transmitter for an exhibition for the lab.

Having built a successful career and kept relevant by taking my chances on the newest hardware and software, I regard this interest in vacuum tubes as decadent.

I do like antiques, provided they

ADVANCED TECHNOLOGY

■ Project leader, Tom McGuire and a prototype compact fusion generator.

Fusion Within 10 Years?

In a way, it's a curious thing that most of the discussions on alternative energy are focused on things like solar and wind power when the obvious long-term solution is nuclear fusion. For half a century or so, fusion research has produced a string of apparent breakthroughs that went nowhere and promises that were not kept, so considerable skepticism has set in. However, the latest — and likely boldest — pronouncement that fusion power is imminent comes from none other than the legendary Lockheed Martin (**www.lockheedmartin.com**) Skunk Works — a mysterious group that has generated advanced aircraft designs since 1943. The organization also has a 60 year history of fusion research and investment, so we're not talking about a bunch of grad students in a University of Podunk lab. These researchers are confident that they will have a working prototype of a compact fusion reactor (CFR) within five years, and a developed and deployed product in 10.

"Our compact fusion concept combines several alternative magnetic confinement approaches, taking the best parts of each, and offers a 90 percent size reduction over previous concepts," explained Tom McGuire, compact fusion leader for the Skunk Works' Revolutionary Technology Programs. "The smaller size will allow us to design, build, and test the CFR in less than a year." After five cycles of testing and refinement, the prototype should be ready.

Yes, many of us feel a little like Charlie Brown after Lucy snatched the football away for the hundredth time. Still, it's great to imagine ships and aircraft that can operate for years without refueling, spacecraft that can get to Mars in only a month, and a power plant that will fit on a semi-trailer and generate 100 MW (enough to power 80,000 homes) — all on less than 55 lb (25 kg) of fuel. So, what the heck. Hold the ball for me one more time, Lucy. ▲

World's Record Data Transmission

Most users don't really think about it, but watching a video on a smartphone or tablet involves a complicated series of transmitters, antennas, receivers, and so forth. With an increasing number of users in the system, a demand for ever-higher image resolution, and a limited number of available frequency bands, it becomes increasingly difficult to handle the huge amounts of data. One possible solution is to move up from the megahertz bands to frequencies of 100 GHz and above which would provide access to a larger band of empty frequencies and higher data rates. Many researchers around the world are working to design circuits that can transmit strong enough signals at these higher frequencies, and now a Swedish group from Chalmers University of Technology (**www.chalmers.se**) and Ericsson (**www.ericsson.com**) has reported success in setting a new world's record transmission rate.

"We have designed circuits for signals at 140 GHz, where we have a large bandwidth. In laboratory testing, we have achieved a transmission rate of 40 GB of data per second, which is twice as fast as the previous world record at a comparable frequency," according to Herbert Zirath, a Chalmers professor and part-timer at Ericsson Research. The breakthrough is the result of materials development that enabled the production of advanced indium phosphide circuits which are so small that details can only be distinguished by using a microscope.

Envisioned applications include high speed transmission of cultural and sports events and communications between computers and data that's stored in the "cloud." This can result in fewer cords and smaller antennas, as well. Ericsson is particularly interested in using the circuits for transmissions between base stations and cell towers. The goal within the project is to demonstrate wireless data transfer of 100 Gbps within the next few years. ▲
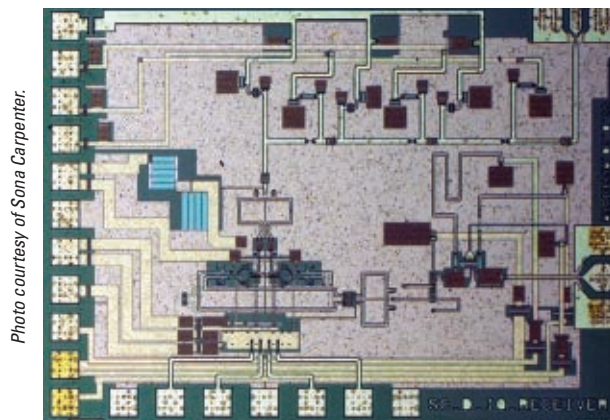
Photo courtesy of Sona Carpenter.

■ A 1.6 x 1.2 mm 140 GHz transmitter chip with I-Q modulator, three-stage amplifier, and a 3x frequency multiplier.

COMPUTERS and NETWORKING

High-End Convertible from Lenovo

You may be more familiar with Lenovo's (**www.lenovo.com**) low-end laptops, which start as cheap as $250 if you can put up with a Celeron processor. However, the company also offers some loftier machines, and one of the latest is the YOGA 3 Pro convertible. Lenovo says it operates in four different modes (laptop, stand, tent, or tablet), but whether the last three are truly different is debatable. "Configurations" might be a better term.

■ Using a watchband-style hinge, the YOGA 3 lays flat.

In any event, you get a 3200x1800 13.3 in display and Intel's new Core M processor which drains just 4.5W. Even so, the YOGA 3 still has an internal fan, presumably mostly to prevent overheating when it's running in Turbo Boost. Even in the overclocking mode, performance reviews tend to be so-so.

For example, a Mobilgeek Cinebench test showed that the cheaper Surface Pro 3 with a Core i5 gives you about 50 percent more CPU power. On the positive side, the design provides some nice features such as a watchband-style hinge that allows it to be thinner (12.8 mm, or 0.5 in) and more flexible. The hinge is made up of more than 800 pieces of steel and aluminum, though, so it remains to be seen how durable it is.

The machine also includes Harmony adaptive software that allows the user to customize how applications behave. For example, when reading an e-book, Harmony will adapt the brightness and color to suit the environmental lighting. It can even optimize audio settings when you're watching a video and play the sound through its JBL speakers. All very nice, but is it worth $1,300? It's your call. ▲

Real Time Translator

In case you didn't notice, Skype Technologies — formerly owned by eBay — was purchased by Microsoft (**www.microsoft.com**) in 2011, allowing it to phase out the Windows Live Messenger service. The big news is that Microsoft has unveiled Skype Translator: a feature allowing people who speak different languages to communicate in (more or less) real time over the Skype network.

A couple of years ago, the company announced that using a technique dubbed Deep Neural Networks, researchers had made a breakthrough in speech recognition that greatly reduced the error rate, and this has allowed for better automatic translation. It isn't clear how many languages this beta version is (or will be) able to translate, but it was demonstrated in English and German, and Microsoft plans to get it working with "a number of other languages."

The Germans in the audience were generally of the opinion that the translation was "not so good" but mostly understandable. You can, of course, see a video of the demonstration on YouTube.

It is not clear at this point whether it will be a free feature or users will have to pay for it, but either way, it may not be long until we can link up with people all over the world and insult them in their own languages.

▲

CIRCUITS and DEVICES

New Flavor of Pi

The Raspberry Pi educational computer has been around since February 2012, which is long enough for some imitators to arrive on the scene. The latest is the Banana Pi from China. The similarities are fairly obvious, but the Banana Pi is about 10 percent larger and is powered by Allwinner's A20 chip with a 1 GHz dual-core ARM Cortex-A7, which makes it more powerful. It comes with 1 GB of DDR3 RAM and an ARM Mali400MP2 graphics processor. Plus, its AXP209 power management unit can drive an external HDD without an extra power supply. The Banana Pi can run a variety of operating systems, including Debian Linux, Raspbian, Lubuntu, Arch Linux, Android 4.4, and Fedora. Internet prices seem to cluster around $60. Support is probably still harder to find, but quite a bit of information, downloads, and a user forum can be found at **www.bananapi.org**. ▲

■ The Banana Pi, featuring a dual-core ARM Cortex-A7 processor.

Rotary Encoder Replacement

Conventional rotary encoders — widely used in automotive, industrial, and consumer products — often suffer from reliability problems because of mechanical wear and contamination by grease, dirt, and humidity. As a solution, Austria's ams AG (**www.ams.com**) offers the AS5601 contactless rotary position sensor, designed as a highly reliable replacement that offers a software-compatible incremental quadrature output. In devices that use rotary knobs, the AS5601 and its paired magnet may replace a failed rotary encoder without any change to the host microcontroller or its application software. The device is based on patented Hall position sensing technology, allowing it to perform contactless rotary position measurement while avoiding the usual reliability pitfalls. It is also designed to reject interference from external magnetic fields. According to

ams, "The AS5601's operation — including its zero position — provides for easy configuration since its register settings are accessed via an I²C interface and are saved in on-chip OTP memory. In addition, the quadrature (A/B) output offers great flexibility, providing between eight and 2,048 positions. This means that the AS5601 may be used, for instance, by off-the-shelf rotary knob or encoder manufacturers, in multiple end products with different output requirements. Users of the AS5601 also have the choice of a 12-bit digital output, suitable for designs that are not directly replacing a conventional rotary encoder." It was also noted that the device includes a pushbutton function and, by default, it automatically enters one of three low-power modes to minimize power consumption. In its lowest power mode, it draws only 1.5 mA. ▲
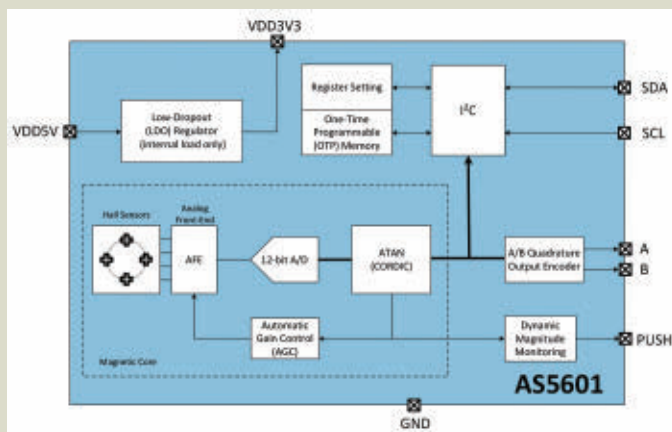
■ Contactless position sensor offers greater reliability.

CIRCUITS and DEVICES Continued

Annoy Your Neighbors!

This month's "got to have" device is the Ninja Remote Stealth Television Gadget and IR Jammer, available from the strange folks at ThinkGeek, Inc. (**www.thinkgeek.com**). This is a high-tech trolling device resembling a standard remote control, but it allows you to "annoy the crap out of friends and enemies by possessing their TV. Perfect for interrupting stupid sporting events, obnoxious video games, moronic sitcoms, idiotic talking heads, and poorly scripted movies."

■ The Ninja Remote Stealth Television Gadget and IR Jammer.

Obviously, it also can provide tons of fun in bars and restaurants, doctor's offices, department stores, and other locations. The Ninja is said to control virtually any TV or digital camera from distances of up to 400 feet. In addition to standard control functions, the bomb button causes the TV to randomly change channels and volume for five to 15 minutes, and the jam button blocks other remotes so your victim remains helpless. Plus, the digital camera shutter button lets you take selfies from a distance. As of this writing, it is sale priced at only $14.95. ▲

INDUSTRY and the PROFESSION

Government Requests Report

If you are concerned about the US government poking into your Internet activities (or especially if you aren't), it might be worthwhile to check out Facebook's Government Requests Report — the third edition of which was released in November. Facebook's deputy general counsel, Chris Sonderby, noted, "Since our first report, we've seen an increase in government requests for data and for content restrictions. In the first six months of 2014, governments around the world made 34,946 requests for data — an increase of about 24 percent since the last half of 2013. During the same time, the amount of content restricted because of local laws increased about 19 percent."

Facebook says that it looks at every government request to make sure that it complies with the strict letter of the law, and it pushes back against overly broad requests such as a New York court's demand for data from the accounts of nearly 400 people. Facebook complies with the vast majority of requests, as documented in the report. To take a look at it, just navigate to **https://govtrequests.facebook.com**. **NV**

Q & A

■ WITH TIM BROWN

In this column, Tim answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. **Send all questions and comments to: Q&A@nutsvolts.com.**

• # Size of Satellite Dish
• # Lump in Power Cable
• # Mailbag

Post comments on this article at **www.nutsvolts.com/index.php?/magazine/article/february2015_QA**.

Size of Satellite Dish

Q Back in the '70s and '80s, satellite receiver dishes were anywhere from six feet to 18 feet in diameter, but now the dishes are only 18 to 24 inches in diameter. How is this possible?
— **William Jenkins**
Boca Raton, FL

A Short Answer: Improved technology. Long Answer: A satellite receiving "dish" is a parabolic antenna designed to focus the received weak signal at the focal point of the parabola where a Low Noise Block (LNB) down converter lowers the signal frequency to one more efficiently handled by the receiver box electronics and amplifier to boost the signal, while adding the minimum possible noise to the signal for the range of channels used by the transmitters on the satellite.

The transmitting satellite is located in a geosynchronous orbit (stays roughly over the same point on the Earth) approximately 22,300 miles (37,000 km) directly (to the extent possible) above the Earth's equator. The satellite has antennae which receive the signal from an Earth-based station (like the head end facility of cable TV), electronics to convert these signals, and antennae to

MAILBAG

Re: Measuring Current With Clamp-On Ammeter:

Hi Tim! First of all, congrats on landing the Q&A Empire. Always (was/now is again) my first stop when I would peruse *Nuts & Volts*. For you, the sky is the limit in not being tied to one topic per magazine. I really enjoyed the first round in the October issue. There's some discussion I'd like to share and also an extremely useful (and cheap) tool I use to quickly and safely measure AC current. Quick background: I've worked engineering in space/military/gov, 25 years with Hughes Aircraft and currently own an engineering firm consulting with the big firms.

The right-hand rule you showed in your article is great, but confusing. Some mention of conventional current would be helpful since depending on where you were taught, the modern view of current flow is electron flow (as the majority current carrier, except in PNP transistors) where current flows negative to positive (hence, a left-hand rule). Just a thought, not a criticism.

As for my current measuring aid (see **Photo 1**), I bought these very short extensions that are supposed to allow you to plug in multiple wall warts in a power strip. By separating the three wires, it gives you an awesome place to clip your current clamp without any messing with the equipment you're measuring. Also, you can make certain there is no current going through the ground wire. I've made tons of these for friends and colleges. Keep up the great work and I look forward to your future articles.
Steven KJ6STF

Steven, thanks for the feedback. It is good to hear
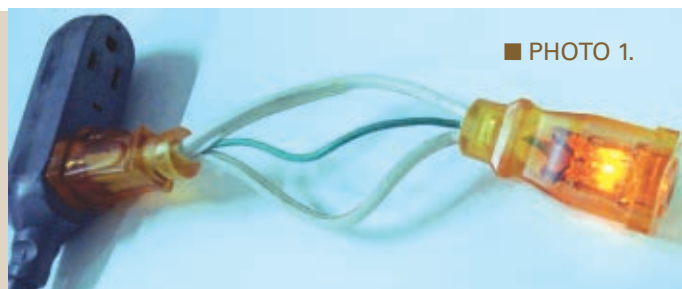
■ PHOTO 1.

from our experienced readers whose job is to keep me straight. As far as not calling out conventional or electron-flow current, I did not think it was necessary to explain why you need to clamp an amp probe on only one wire at a time due to the effective cancellation of magnetic fields between the two conductors. Plus, when you open the conventional versus electron-flow "can of worms," you polarize your readers who each think one or the other is best (I think BOTH are best because each has its use in explaining electrical charge flow), and there is the need to take extra space to explain the two systems for those who are new to electronics.

I like your idea for a "tool" to aid in measuring current with an amp probe. I would just remind readers to remember to use this device ONLY within its voltage and current limits. Avoid exposed wiring or other conductors which could be lethal when touched.

I have the easy job at N&V because all I have to do is come up with answers. The authors of the other columns have to come up with the questions (project ideas) and the answers too. Our readers have the challenge to keep my answers straight.

12 NUTS&VOLTS February 2015

transmit the satellite TV signal to Earth over a broad geographic area called the "footprint."

In the 1970s and 1980s, satellite TV used the C-band which covers frequencies 4 GHz to 8 GHz (1 GHz = 1 billion cycles per second). In the 1990s, the satellite TV downlink signals were switched to the Ku-band which covers the frequency range from 12 GHz to 18 GHz. One of the dictates of antenna design in essence says the higher the frequency, the smaller the antenna needed. This is one of the reasons the modern satellite receiver dishes can be made smaller.

Another reason for smaller dishes (hope I don't sound like a rapid weight loss commercial) is the improvements in the solar panels and electronic devices — particularly, the power amplifiers on the satellite which allowed them to transmit a higher power signal (less than 10 watts in the early days versus 100 watts today). The more transmitted power, the smaller the receiving antenna that is needed to capture sufficient signal to give a necessary signal-to-noise ratio.

I failed to mention that with the LNB, the electronics in today's satellite dishes produce a lot less noise than those in the earlier years. All of these factors — improved electronics, higher transmission frequencies, and higher

transmitter power — work together to greatly reduce the size of the satellite receiving dish. (There were also improvements in satellite-station keeping, which meant the transmitted signal had less "overspray" which reduced the signal level at the dish. However, that is rocket science from which I will spare you.)

I hope this answers your question without overpowering (pun intended) your brain cells.

Lump in Power Cable

Q I have noticed that computer power cords have a lump at the end. What does this lump do and do I need it on my computer cord?

— Tom Dixon
Billerica, MA

A I'm sure that many people are wondering the same thing. This "lump" on the end of computer (and other devices) power cables and some USB cables is a ferrite choke (sometimes called a ferrite bead). Ferrite is a material made of iron oxide (rust) and other alloying elements. Ferrite is slightly

---

Re: Comments from November 2014 column
Hi Tim. I have some comments regarding the items in the November publication:

1. **Broken laptop:** The implication is that the laptop won't boot, in which "Waiting for Windows" might be a BIOS screen. Your suggestions are good, but they commence with the implication that Windows is already running. The problem is that if the laptop won't boot, then the writer must use Windows Recovery (if available) which, in turn, can be used to initiate CHKDSK. The laptop startup disk might also contain a copy of Windows Recovery. Best idea would be to take the laptop to someone skilled in the practice.
2. **VCR eats tapes:** I assume that where the writer states "pulling on the tape to remove it ..." means pulling on the cartridge to remove it. I've seen a variation on this problem. I have a vintage 1987 VCR camera which I tried to power up and use several months ago. The mechanism did not work, and a bit of investigation showed that one of the internal drive belts had broken -- after almost 30 years under tension. Of course, I had the advantage of owning a service manual for the machine, so the analysis did not proceed without some foreknowledge and lots of illustrations. Usually, the rubber belts can be replaced by suitably sized o-rings available at a good hardware store.
3. **Sound system hum:** Spot on. Ground loop. My suggestion would be to disconnect every equipment interconnection (keeping good notes, if needed). Reconnect one at a time, beginning with the

amplifier-to-speaker connections, until the hum starts. Note this and leave the offending cable disconnected. Continue this process until everything that does not cause hum is connected. This leaves some specific offenders to investigate and, as you suggested, isolate.

Peter A. Goodwin, Rockport, MA

Peter, thanks for the comments and added information. I will be the first to admit that I don't know everything (sometimes I wonder if I know anything), so our readers are a vast resource for problem-solving information. I have to say I'm impressed that you have a 27 year old VCR still working. I have bought newer devices usually to get the benefits of upgraded features and functions (I just replaced a 13 year old computer that was beyond hope — third hard drive failed, failing video adapter, and swelling motherboard capacitors, plus old and slow processor).

I failed to mention that getting the laptop open is somewhere between a fine art and magic, so it is imperative that you have manufacturer's instructions to avoid damaging the case and components.

Troubleshooting noise in sound systems (as does any device) requires a lot of analysis before taking major "surgical" actions. I always recommend as a first step to find out any changes that have been made to the system and when those changes were made (most PA system users don't keep maintenance logs or up-to-date schematics). Then, start checking mic connections, power line, devices inputting to the sound board, and sometimes devices receiving outputs from the sound board.

magnetizable, and your "lump" has been fabricated into a hollow cylinder that will fit around the cable and be covered with plastic (molded on or attached).

The purpose of the ferrite choke is to prevent electrical signals from passing easily. In the case of the computer, these signals are generated by the computer's clock circuitry and try to leave via the power cable; they cause interference with other devices connected to the AC power lines. Radio frequency (RF) signals may also try to enter your computer via the power cable.

The theory behind this is based on Faraday's Law (discovered by Michael Faraday, 1791- 1867) which in essence says, "A magnetic field moving across a conductor (wire) will generate an electrical voltage (this is how generators work), and likewise a changing voltage across a conductor will cause a changing current which will, in turn, create a changing magnetic field around the conductor."

[Side Tour: James Clerk Maxwell (1831-1879) discovered the equations which bear his name and extend Faraday's Law to quantify radio wave propagation.]

Thus, any electrical signal passing through the air such as radio waves can and do induce a small signal in any conductor they cross. These "unwanted" signals are considered as interference (similar to static on the AM radio for those old enough to remember). The magnetic properties of the ferrite surrounding the power wires prevent most of the interfering signal from passing and causing trouble (high inductive reactance is the technical term). So, unless you want to take a chance with receiving or transmitting interference signals, you need the ferrite.
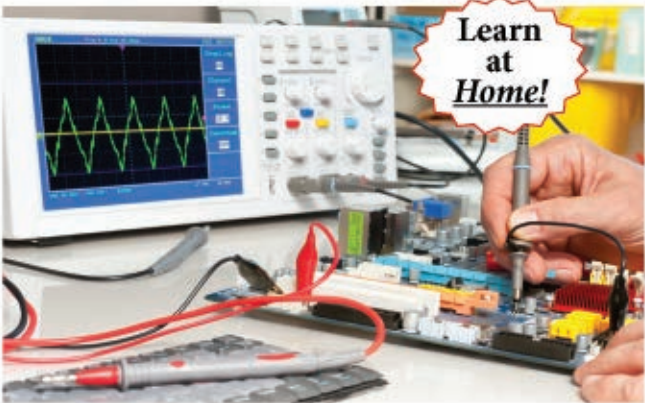
For anyone wanting to learn more about radio frequency signals and propagation, *N&V* columnist Louis Frenzel has a number of articles which should be of interest. Plus, as always, the Internet is rife with resources and information. **NV**

Load and Go!

I've written a fair number of Propeller applications and — until recently — never had a problem fitting one into 32K of memory. Last year, my friend John and I (EFX-TEK) began work with a gaming company to develop a new controller for one of their products. Moving from a small PIC to the Propeller gave them the opportunity to add exciting new features. In the end, though, we just couldn't fit everything into one application. On analysis, what they wanted was, in fact, three discrete applications for the board. Moving between these apps would be infrequent, and didn't need to be instantaneous. As the product includes a microSD card, I developed a bit of code to load the pre-compiled apps from it.

If you're very new to the Propeller, you may not have considered its architecture vis-à-vis other micros. In a Propeller-based system, the program code is stored on an external 32K EEPROM (it can be larger, but only 32K is used for program space). On reset, the Propeller will copy the contents of the EEPROM to internal RAM for execution. The up-side of this architecture is that we can modify the EEPROM with the current program to affect things the next time it reboots. On the extreme end of that scenario, we can replace the code in the EEPROM with a completely new application.

In previous projects we've used SD card access (*fsrw.spin*) to read files, and an EEPROM object (*jm_24xx256.spin*) to read and modify the boot EEPROM. The code presented here will combine these elements to simulate our project being reprogrammed from an IDE (integrated development environment), though it will happen in place — without a computer connection — and on demand. As was the case for our customer, this will allow us to selectively run multiple applications that wouldn't otherwise fit into 32K if combined on a Propeller board that has an SD interface.

Propeller Image Files

Propeller IDEs allow us to download to RAM, or to RAM and EEPROM. How does this work? When a download request is made by the IDE, the Propeller runs a bit of boot loader code that is embedded in the silicon. The boot loader facilitates moving the code from the IDE through the serial link and into the Propeller's RAM. If the EEPROM update is part of the request, the Propeller will write what's in RAM to the EEPROM.

If you use the Propeller Tool (Windows only), pressing F8 will display the Object Info window as in **Figure 1**. This displays the compilation details and allows us to save the file in BINARY or EEPROM format. Why two formats?

The EEPROM format creates a 32K image file that can be used by EEPROM programmers; this allows a high-volume manufacturer to preload code into the EEPROM before it is soldered to the PCB (printed circuit board). We could, in fact, copy an EEPROM file from the SD card to the Propeller's EEPROM and reboot — I've done this and it works. That said, most applications don't require the entire 32K; in fact, most of the 32K is available for variables and the stack.

The BINARY option creates a
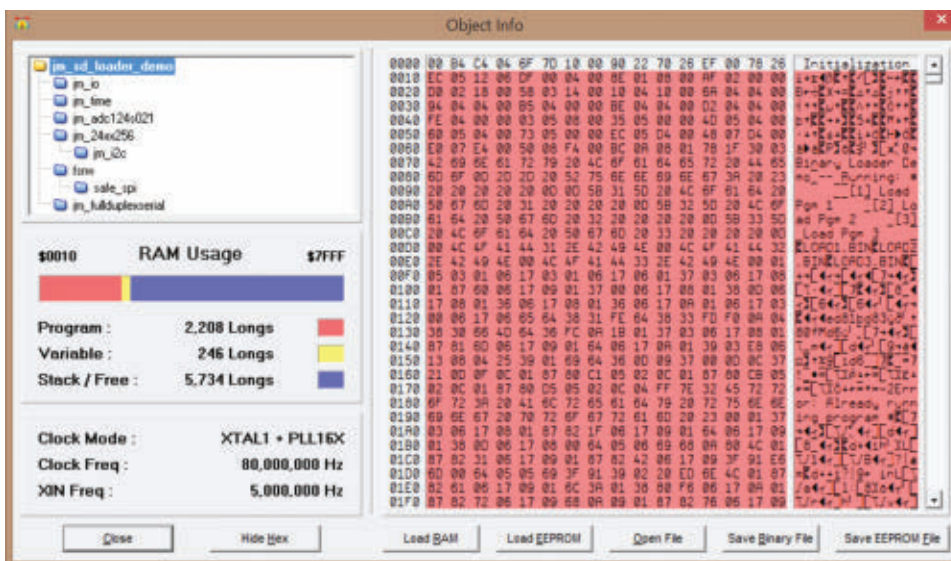
■ FIGURE 1. Object information.

Post comments on this article and find any associated files and/or downloads at
www.nutsvolts.com/index.php?/magazine/article/february2015_SpinZone.

ITEM	DESC.	SOURCE/PART #	
Prototype			**BOM**
Propeller Activity board		Parallax 32910	
uSD socket		Parallax 32312	
DIY (SMD)			
C1	0.1 µF 0805	Mouser 77-VJ0805Y104MXXAC	
R1..R5	10K 0805	Mouser 71-CRCW0805-10K-E3	
SKT1	uSD SMD	Mouser 798-DM3CSSF	

file that is just the compiled code and data. This image also includes information about the variable space and start of the main program stack; this information is provided in the first 16 bytes of the file. Note that on the Object Info dialog, the ASCII display shows "Initialization."

Let's have a look at the details of those 16 bytes (multi-byte values are stored Little Endian):

0..3	(Long) Clock frequency
4	(Byte) Clock mode
5	(Byte) Checksum
6..7	(Word) Start of code/data
8..9	(Word) Start of variables
10..11	(Word) Start of stack
12..13	(Word) Current program pointer
14..15	(Word) Current stack pointer

For the purposes of loading a binary image into EEPROM, we're going to be using the checksum byte (for validation), the start of variables (called VBASE), and the start of the stack (called DBASE).

If you've looked at the EEPROM map, you can see that the variables and stack space are initially cleared to zero. As this is the case, the IDE only downloads the compiled program code (which includes DAT tables). If you look at an EEPROM file from Propeller Tool, it's always 32K in size because it's a byte-for-byte map. A BINARY file, on the other hand, is always smaller because it only contains the compiled code and data.

Once the IDE downloads the compiled program image, the boot loader takes care of clearing the variable and stack space (which is everything from the end of the compiled code to the end of the 32K EEPROM space). A final step is to set the initial stack frame values. Once that is set up, cog 0 is loaded with a Spin interpreter and pointed to the compiled Spin code (which always starts at $0010, just after the initialization data).

Validating the Binary Image

The SD card connected to our project could contain any number of files, and some of them could have the correct extension (.BIN or .BINARY). We can use the checksum byte and the start of variables value to perform

a simple validation of the target file. Byte 5 of the initialization header is the checksum which is derived by adding all the bytes in the binary image, plus the initial stack frame values (two longs of $FFF9_FFFF). If all is well, the low byte of the sum will be $00. Note that the checksum calculation is normally done after the Propeller has loaded the binary image, cleared the rest of RAM, and then set the initial stack frame.

Let's jump in with a method that will validate a file before moving it into the EEPROM. This is an important first step so that we don't "brick" (crash) our project with a bogus file:

```
pub check_binary(p_filename) {
} | fs, check, cs, p_buf

  ifnot (has_file(p_filename))
    return false

  sd.popen(p_filename, "r")
  fs := sd.get_filesize

  sd.pread(@buf1, 16)
  check := buf1.word[4]
  if (check <> fs)
    return false

  cs := $EC
  sd.popen(p_filename, "r")

  repeat
    check := sd.pread(@buf1, BUF_SIZE)
    if (check =< 0)
      return (cs.byte[0] == $00)
    else
      p_buf := @buf1
      repeat check
        cs += byte[p_buf++]
```

The **check_binary()** method takes a string pointer for the name of the file. As FSRW uses DOS 8.3 naming conventions, we should talk about that first.

The Propeller Tool and PropellerIDE save compiled binary images with a .BINARY extension. For my own projects, I tend to change the extension to .BIN which allows me to use all eight characters for the file name in my listings. If you prefer to keep the .BINARY extension, you'll need to limit the number of characters in the name to six, and then add ~1 before the extension in the code. This is to say that if the file is called LOAD1.BINARY on the SD card, in the program we must refer to it as LOAD1~1.BIN. My preference is to rename the file so that its name on the SD card matches the listing. It's always best to keep it simple.

If the file exists, we open it and read the file size into variable *fs*. The next step is to read the first 16 bytes of the file into a buffer. As the file was just opened, this will be from the front end of the file. We can extract Word 4 from the buffer which is the RAM address where the variables start. Remember that the file only contains compiled code and data, hence the value at Word 4 should match the file

size — this is our first validation step.

If the file size is validated, we can run the checksum calculation on its contents. Note that we must preset the checksum byte to $EC to account for the initial stack frame values (two longs of $FFF9_FFFF) which are not included in the file. To simplify the checksum loop, we'll re-open the file to reset the FSRW file pointer to the beginning.

At the top of the checksum loop, we read a block of the file into a RAM buffer. The buffer size has been set to the size of an EEPROM page to facilitate writing later. Remember that the **.pread()** method will return the number of bytes read (now in *check*); if this value is 0 or negative, we are done with the file. If the value is greater than 0, we iterate through the buffer adding each byte into the checksum. When the end of the file is reached, we will return **true** if the low byte is $00; **false** if not.

Remember that the point of checking the file on the SD card is to ensure that we don't load something that will brick our project and require a computer to correct. Long term, the customer John and I are working with wants to send new program images over an XBee radio that is part of his product. After receiving the new file over the air, the **check_binary()** method can be used to validate the download.

■ **FIGURE 2. microSD interface.**
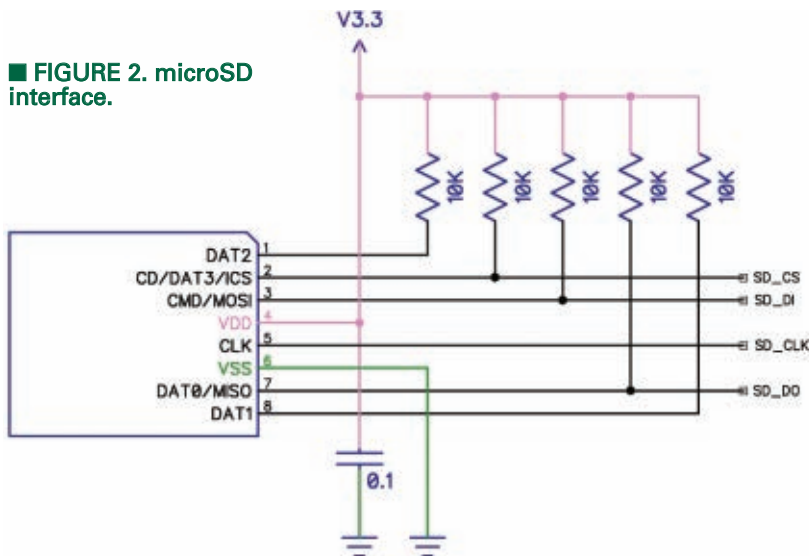
Loading the EEPROM

The process of moving the binary image to EEPROM is fairly straightforward, and actually mimics the download process from the IDE: 1) Copy binary image to EEPROM; 2) Clear the rest of EEPROM to zeroes; 3) Set the initial stack frame values; and 4) Verify the checksum.

```
pub load_eeprom(p_filename) {
} | dbase, check, eeaddr, cs, p_buf

  sd.popen(p_filename, "r")
  sd.pread(@buf1, 16)
  dbase := buf1.word[5]

  sd.popen(p_filename, "r")
  eeaddr := $0000

  repeat
    longfill(@buf1, 0,  BUF_SIZE >> 2)
    check := \sd.pread(@buf1, BUF_SIZE)
    ee.wr_block(eeaddr, BUF_SIZE, @buf1)
    eeaddr += BUF_SIZE
    if (check < BUF_SIZE)
      quit

  longfill(@buf1, 0,  BUF_SIZE >> 2)
  repeat while (eeaddr < $8000)
    ee.wr_block(eeaddr, BUF_SIZE, @buf1)
    eeaddr += BUF_SIZE

  ee.wr_long(dbase-8, $FFF9_FFFF)
  ee.wr_long(dbase-4, $FFF9_FFFF)

  eeaddr := $0000
  cs := $00

  repeat while (eeaddr < $8000)
    ee.rd_block(eeaddr, BUF_SIZE, @buf2)
    p_buf := @buf2
    repeat BUF_SIZE
      cs += byte[p_buf++]
    eeaddr += BUF_SIZE

  return (cs.byte[0] == $00)
```

We start by reading the initialization values into a buffer, then capturing Word 5 into a variable called *dbase*; this is the address where the variables end and the stack (free RAM) starts. Reopening the file resets the file pointer to the beginning. The EEPROM address is preset to $0000.

A simple **repeat** loop moves the file into the EEPROM. Before reading from the file, the buffer is cleared to zeroes. The reason for this is that the last read from the file will probably not fill the buffer; as we're writing the entire buffer into the EEPROM, we need to ensure the buffer bytes not affected by **.pread()** are zero.

With the file contents loaded, we clear the rest of the EEPROM to zeroes — just as the boot loader does. The third step is to write the initial stack frame values into EEPROM; this is required for returns and **abort** traps to work

correctly. The final step uses a loop to read back the contents of the 32K EEPROM and validate the checksum.

If **load_eeprom()** returns **true**, we can use **reboot** to start the new program — at this point, we have only changed the EEPROM, not what is currently running. We must reboot to run the new code. You're probably wondering, "*Okay, how long does this take?*" About 25-ish seconds at 80 MHz. Yes, it seems like a long time when watching it, but in practice, it's really not bad.

Using a PASM I²C driver for the EEPROM will speed things up quite a bit. I didn't do that with the customer project because it consumes all eight cogs. Down the road, we'll probably revisit this; it's certainly possible to shut down another cog to make room for a fast I²C cog, but that's a project for another time.

Putting It Together

In the customer application, there is a constant called PGM_ID. This is 1 for player mode, 2 for configuration mode, or 3 for referee mode. When the program first starts, it looks at a set of inputs to determine which application is desired. If the requested app and the current app match, no action is required and the program continues running.

If the request differs from the current app, we run through the process to verify and load the file for the new application. This probably goes without saying, but each of the applications requires the ability to read files and write to the EEPROM. As is my habit, I created a template file for the customer project that handles the EE loading; from this base template, new applications are created.

I've included a demo app with a template built around the Propeller Activity board which includes a microSD socket. If you're working with a breadboard-based system, Parallax has a microSD adapter (see **Resources**); if you're going to roll your own, **Figure 2** shows the circuit I use in my projects.

In the demo, we use a terminal for user input. When the program is running, it will present a menu as shown in **Figure 3**. When a new program is selected, we use the processes described above to verify and load the new image — the output will look something like **Figure 4**.

Embedded Code

Last summer, I helped my friend Ryan (DEFCON) and Parallax with code for the DEFCON 22 badge. Once the code was approved, I worked with David Carrier at Parallax to create a test program that would exercise the
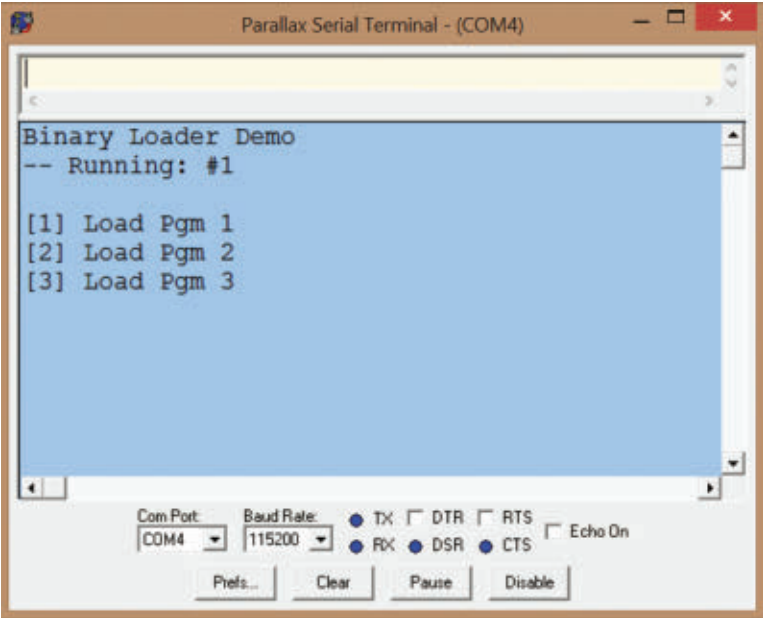
■ **FIGURE 3. Loader menu.**
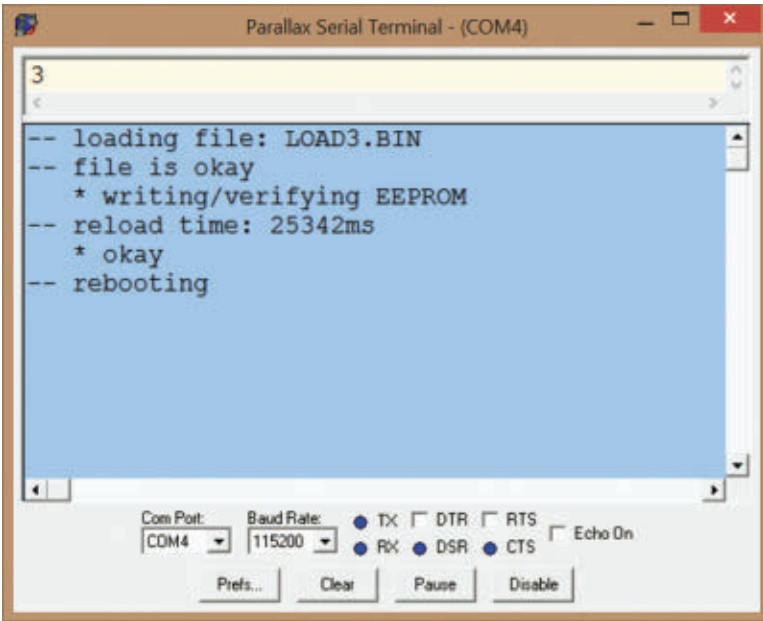
■ **FIGURE 4. Good reload.**

badge hardware. If the hardware test passed, the program would load the embedded code.

"*How do I embed code?*" you wonder ... it's pretty easy. In a DAT table, we use the FILE type instead of BYTE, WORD, or LONG like this:

```
DC22        file      "final.binary"
DC22x       byte      0
DC22pad     byte      0[BUF_SIZE]
```

Again, the compiled image must be small enough to fit into the existing application — the compiler will
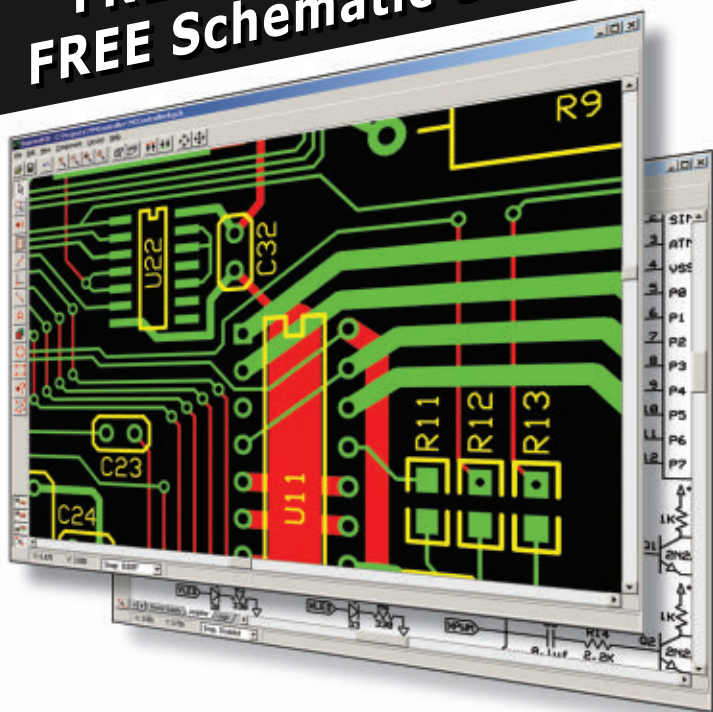
complain if it's too big. The reason for the label immediately after the file data allows us to calculate the length of that data. Finally, I include a pad that is the same size as the buffer; this ensures that I can do block writes of the EE page size which allows the process to run as fast as possible.

Neither the badge hardware test nor the badge code was very big, so this was the approach we took; if the test passed, the tester would allow the badge to reprogram itself from the embedded binary image. The process is identical to what I described above, except that the program image is copied from RAM to the EEPROM, and no SD card is required.

Advanced SD Loading

If the idea of loading compiled Propeller code piqued your interest, you may want to check out an application called Spinix by Dave Hein. It's actually a simple operating system for the Propeller, inspired in part by Linux. I've played with it a bit and it's quite interesting. Dave is a gifted programmer and hero contributor to the Propeller forums, so his work is well worth looking at.

Tricks

You may have noticed that I defined the RAM buffers as longs, even though the EEPROM is byte oriented. The reason for this is long alignment allows us to use the **.word** and **.byte** modifiers with the buffer name (if a variable is long aligned, we can access it as words and bytes). This is a subtle trick that you will see in some Spin programs — watch for it!

Until next time, keep spinning and winning with the Propeller! **NV**

NEW PRODUCTS

■ HARDWARE
■ SOFTWARE
■ GADGETS
■ TOOLS

PARALLEL GRIPPER KIT

The Parallel Gripper Kit A from ServoCity is a simple, durable, and versatile kit that is perfect for all kinds of robotic applications. The grippers move towards one another in a parallel motion — hence, the name — and have a maximum width of 2.80". By incorporating the Actobotics 0.770" hub pattern into the back plate and 1.0607" hole spacing on the grippers, users can easily attach nearly any Actobotics component to the assembled kit. The kit is designed for use with any standard size Hitec or Futaba servo (sold separately), and is easy to assemble. It only requires a Phillips head screwdriver. Price is $14.99 ea.

HORIZONTAL & PERPENDICULAR GRIPPER KITS

Also new from ServoCity in the Actobotics gripper kit line-up is the Horizontal Standard Gripper Kit A and Perpendicular Kit B. The sleek and simple design makes assembly and attachment very easy. The grippers have a maximum width of 4.20" and contoured design to help hold on to larger objects. The

Horizontal kit incorporates a 0.770" hub pattern into the back plate that can be easily attached to nearly any Actobotics component. The Perpendicular kit also offers many mounting options and is designed for use with any standard size Hitec or Futaba servo (sold separately), as well. This kit is equally easy to assemble with just a Phillips screwdriver. Pricing is $9.99 ea.

For more information, contact:
ServoCity
Web: **www.servocity.com**

SBC WITH 1 GHz ARM CPU, eMMC SLC FLASH, AND HIGH CAPACITY FPGA

Technologic Systems, Inc., has announced the availability of their newest industrial grade single-board computer: the TS-7250-V2. The TS-7250-V2 is a general-purpose PC/104 embedded system that can fulfill a wide variety of embedded system requirements with high performance, high reliability, low cost components like an 800 MHz or 1 GHz ARM CPU, 512 MB DDR3 RAM, up to 17k LUT FPGA, and 2 GB eMMC SLC Flash, and a flexible PC/104 connector with FPGA driven
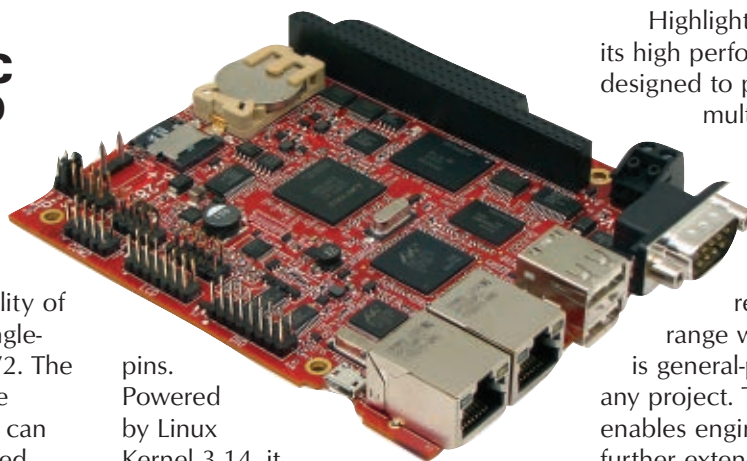
pins. Powered by Linux Kernel 3.14, it boots to a terminal shell in 0.87 seconds and can further load the Debian 7 "Wheezy" distribution. The TS-7250-V2 provides an upgrade path for customers using the TS-7250 or TS-7260 single-board computers.

Highlights of the TS-7250-V2 are its high performance components designed to provide powerful multimedia capabilities and industry standard interfaces, while maintaining low power consumption and high data reliability. Applications range widely as the TS-7250-V2 is general-purpose and fits nearly any project. The high capacity FPGA enables engineers to customize and further extend the TS-7250-V2 to meet their application specific needs. Furthermore, the industrial temperature option enables the TS-7250-V2 to function in rugged environments such as those found in the automotive and military industries.

Some of the hardware features include:

- 1 GHz ARM Marvell PXA168
- 800 MHz ARM Marvell PXA166
- 512 MB DDR2 RAM
- 2 GB eMMC SLC Flash Storage
- 8 or 17k LUT Programmable FPGA
- 2x 10/100 Ethernet Ports
- -40°C to 85°C Industrial Temp Range
- 1x microSD Card Slot
- 1x SD Card Slot
- 2x USB Host Ports
- 1x USB Device Port (console only)
- 75x Digital Input/Outputs
- 3x RS-232 Serial Ports
- 3x TTL Serial Ports
- 1x RS-485 Serial Port
- 5x ADC Ports
- 1x CAN Bus
- 5 VDC Power Supply Input
- 8 to 28 VDC Power Supply Input
- High Precision Real Time Clock

The TS-7250-V2 also features an onboard programmable Lattice FPGA. The FPGA provides extra peripherals such as 75 GPIO lines, additional serial ports, additional CAN port, dual four-bit mode high speed SD card sockets (full size and microSD cards), and more.

All the PC/104 pins are connected straight to the FPGA, giving the TS-7250-V2 the flexibility to add external hardware and physical/transceiver layers. The default FPGA load provides a standard PC/104 bus on the 104-pin connectors, maintaining compatibility with other ARM SBCs and a wide range of PC/104 peripheral boards.

Technologic Systems also includes user space utilities for controlling various aspects of the board, like DIO, SPI, and UARTs.

Some of the software features include:

- Linux Kernel 3.14 (2.6.34 also available)

- Debian 7 Wheezy Distribution
- 0.87 sec Bootup Time to Busybox Terminal Shell (.52 seconds with available 2.6.34 kernel)
- FPGA Reloadable in Software
- Flexible Programming Language Support
- Source Code Examples Available

The TS-7250-V2 single-board computer is available in quantity 100 pricing starting at $165 with several additional options available.

The TS-7250-V2-512-8S-I: TS-7250-V2 industrial grade (-40°C to 85°C) SBC with the Marvell PXA166 800 MHz ARM9 and 512 MB DDR2 RAM (microSD required to boot) is priced at $199 (quantity 1).

The TS-7250-V2-512-4096F-10S-I-DEV: TS-7250-V2 industrial grade (-40°C to 85°C) SBC with the Marvell PXA168 1066 MHz ARM9, 512 MB DDR2 RAM, 2 GB Flash (4 GB eMMC configured as 2 GB SLC Flash), 2x 10/100 Ethernet, and 1x RS-422 (recommended for development) is priced at $251 (quantity 1).

HumPRO WIRELESS DATA MODULE FOR INTERNET OF THINGS

A s the fast-emerging Internet of Things (IoT) continues to gain momentum, the demand for reliable low cost wireless data modules increases every day. In response to this demand, Linx Technologies has announced the release of its low cost 900 MHz HumPRO™ series wireless

data transceiver module.

The HumPRO 900 MHz version outputs up to 10 dBm. This results in a line-of-sight range of up to 1,600 m (1.0 miles). Other sub 1 GHz frequencies will be added in the near future.

The HumPRO series is a Frequency Hopping Spread Spectrum (FHSS) transceiver designed for the reliable transfer of digital data. Built on the Linx Hummingbird platform, the module is below $10 in volume, and is designed for emerging IoT applications where cost is an important driver to adoption.

The ubiquity of low cost Internet connections and development of Internet gateway technology on slim hardware are driving the adoption of the Internet of Things. The cost of the wireless connection from the gateway is a growing concern in IoT implementation — especially in sub 1 GHz applications. At under $10 with FHSS, the HumPRO is a cost-effective wireless solution with a robust connection of over 800 meters in range.

The wireless data module employs a fast-locking FHSS system for noise immunity and higher transmitter output power as allowed by government regulations. The fast-locking module also allows the module to wake up, send data, and go to sleep quickly, making it an ideal solution for battery powered applications. It handles all protocol functions automatically and has options for acknowledgements, Listen Before Talk (LBT), and assured delivery. Multiple hopping patterns enable multiple systems to operate in proximity without interference.

All modules have a unique 32-bit serial number that can be used as an address. Multiple addressing modes give the options for 16- or 32-bit source and destination addressing. The addressing methods support point-to-point and broadcast links. Address masking by the receiving module allows for creating subnets. More advanced network methods can also be implemented with an

external microcontroller.

A standard UART interface is used for module configuration and data transfer. A few simple serial commands are all that are needed for configuration. This is suitable for direct connection to UARTs on many microcontrollers, USB converters, and RS-232 converters.

For more information, contact:
Linx Technologies
Web: **www.linxtechnologies.com**

SLIM OLED
DISPLAYS

Newhaven Display International, Inc., has announced its first line of slim character OLEDs. These new OLEDs bring a more affordable OLED option that is half the thickness and twice the contrast.

Available in 2x16, 2x20, and 4x20 character formats, the new slim OLEDs add additional color options to the OLED display line. Red, green, blue, and yellow colors are offered in each display size. The units come packaged as a complete module that includes all required logic at only a 5 mm thickness. With three built-in font tables, 10,000:1 contrast ratio, 10 μsec response time, and built-in screen savers, these new displays will offer any industry an upgrade to their current monochromatic character application. Other features include:

• 160° viewing angle
• Double height characters
• Built-in US2066 controller
• Modules include all necessary logic

NO-NONSENSE HOMEBREW
ANNUNCIATOR
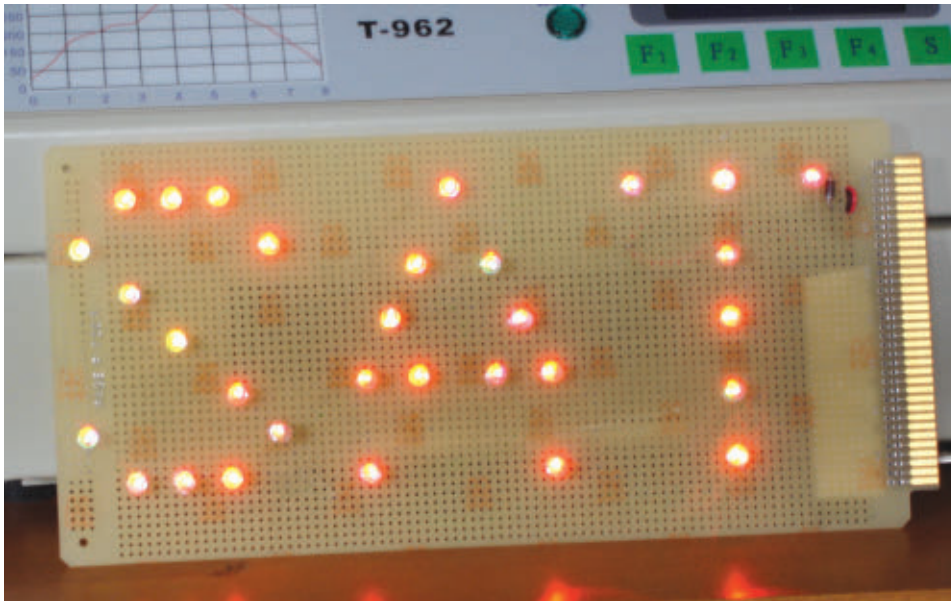
By Ron Newton

■ **FIGURE 1.**
Prototype annuciator.

Having trouble with folks misinterpreting what you're saying? Now, you can be clearly understood with this handy little circuit that lets you "announce" what message you're trying to get out. Plus, I'm calling it "no nonsense" because there is no microprocessor involved. To change the spelling/message on the annunciator, you simply change the position of the LEDs.

Why did I come up with such a project? I'm part of a traditional jazz band called the Carson City Rascals (my former band was called the "Tahoe Toads"). For those who don't know what traditional jazz is, it used to be called Dixieland. Now it's just called Trad Jazz.

I have a two by six foot banner that says "Dixieland Played Here." It is displayed at different jazz clubs that we play in. The problem is that it is usually displayed in a dark alley or a street with dim or no lighting, so it doesn't attract much attention. Also, there is often no power available. So, I ended up using LEDs because they work better for signage.

Since we only play on Saturday nights, I wire wrapped one up using perf board that said "SAT" (**Figure 1**). It is driven by two AA batteries and lasts 24 hours. When asked to play on a Wednesday night, I had to make up a second perf board. This got me thinking about alphanumeric LEDs. I liked the idea of color changing versions that draw far more attention, and they are only about three cents apiece. The other consideration was the sign did not need to be lit in the day time.

I sat down and within a hour had drawn out the positions of 17 LEDs which can spell out any letter or number. These LEDs draw an average of 3 mA using 2.4 volts. (The current varies with the color being displayed.)

The annunciator uses three letters for displaying the days of the week. This amounts to a maximum of 38 LEDs on at one time (MON), or a little over 100 mA. Four alkaline batteries will produce six volts. By turning the LEDs on and off using a transistor, we get an approximate .8 volts diode voltage drop, which is just about perfect.

I added a photoresistor which detects if the sun is up, so it turns off the display during the day. With the transistor base biased controlling its amperage, it ended up about the right voltage for the LEDs without burning them out.

This project is about as simple as you can get. It has all through-hole parts and only requires a soldering iron and a pair of snips. The LEDs plug into headers and can be changed easily for a different word. You can mount several boards together to get more letters.

Building the Board

The board was drawn using ExpressPCB free software which is available at **www.expresspcb.com**. The board files are available at the article link. A kit is available from the *Nuts & Volts* webstore for your convenience.

All the components are placed on the top of the board. Locate the strips of headers and cut six strips of eight headers each. Solder them to the six eight-hole inline pads of the matrix sets on the front of the board.

Cut 27 pairs of headers and solder them to the remaining vacant spots. Solder in the switch, the resistor, and the transistor, noting its polarity. Solder in the photoresistor; it has *no* polarity. Pass the two battery wires from the back of the board through the strain relief hole, then solder the black wire to "B" and the red wire to the red marked pad. Add four batteries to the board and you're done! (Refer to **Figures 2** and **3**.)

Cut all 40 LED leads so they are 3/16" long or just above the flat stamped lead area. The plastic part of the LED has a flat indicating the cathode.

Go to the article link and download the LED matrix for making letters and numbers. Determine which day of the week you want to display and plug LEDs into the headers. The round pad is the cathode and the flats are indicated by board markings.

There is an exception to this if you're making a T, I, or Y. To make a straight line, you will need to turn two LEDs around and reverse the polarity (one for the Y). This is marked in red on the matrix. NOTE: Unlike standard LEDs, these LEDs will conduct if reversed and short out the board. I recommend that you put a piece of tape over the photoresistor, which will turn on the power. Start plugging in the LEDs; if one is reversed, the board will not light (**Figure 4**).
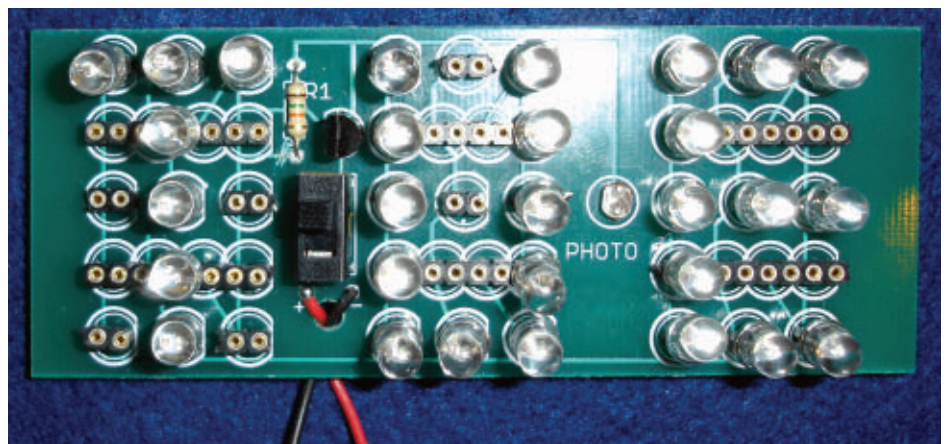
■ **FIGURE 2. Partial board**

■ **FIGURE 3. Stuffed board**

■ **FIGURE 4. Lit board.**

How It Works

This circuit is about as basic as you can get. The NPN transistor provides the voltage drop needed. Six volts positive goes to the collector. R1 (15K) and the photoresistor act as a voltage divider for the base. When light strikes the photoresistor, its resistance is in the area of 10K ohms. When there is no light striking the photoresistor, its resistance is in the area of 100 megohms.

Keep in mind that transistors amplify current (not voltage) and are dependent on their load. The PN2222A has a Hfe somewhere in the area of 100-300, which is the amplification factor. Using 15K ohms at six volts, you have 6 /15,000 = .0004 amps:

$$4^{10-4} \times 200^{hfe} \approx 8.0^{10-2} \text{ or } \approx 80 \text{ milliamps}$$

The battery supply is four alkaline batteries; you determine the size by the amount of days you want it to run. You can also add a five volt battery eliminator if you like. (See Mouser 709-GSM06E05-P1J or Digi-Key T977-P6P-ND.)

Depending on how long you want the unit to run, use these numbers that will give you the approximate on-time:

AAA	11 hours	0.5 days
AA	22 hours	0.9 days
C	66 hours	2.7 days
D	133 hours	5.5 days
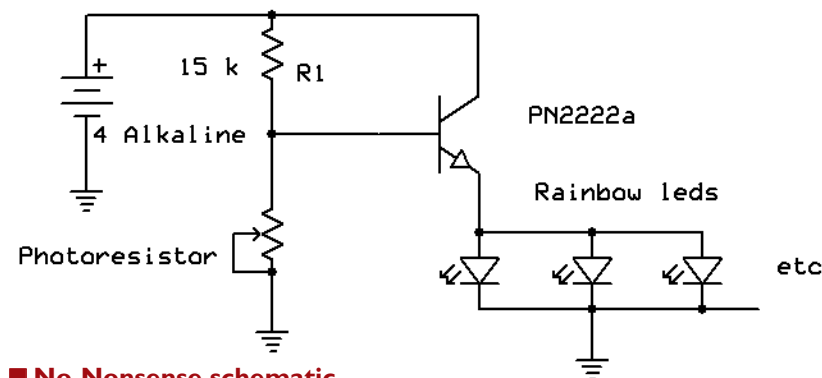
You can double these amounts if you consider 12 hours on/12 hours off with the photodetector.

A Quick Course on Soldering

Here are some basic tools you'll need:

1. Small wire solder with flux (0.60 mm .025").
2. Solder flux (Chip Quik SMD291).
3. Soldering pencil; either 25 or 35 watt. RadioShack has one that lets you can change tips. I do like Weller soldering stations that control the tip temperature. I normally solder with a tip at 700 degrees, and use a long conical tip 1/64" for surface-mount or .03 x 5/8" for through-hole.
4. A pair of nippers (RS6400064 at RadioShack).
5. Solder braid .030 for surface-mount and small pads; .060 for general use.

I have developed the habit of using a small amount of solder flux on each pad. It saves a lot of grief.

1. Get in the habit of cleaning the tip each time with a wet sponge to remove excess solder and flux.
2. Place the soldering tip on the component lead and the pad of the board, and heat both. Add the solder to the lead and pad, but not to the solder tip. Once melted, wick up the solder on the component lead by pulling the solder tip up the component lead. The solder joint should be shiny. Clip off the excess lead.
3. If you have a solder bridge, use solder braid. Add some solder flux to the pads and press the braid against the solder bridge. Heat the braid with the solder tip and allow the solder to flow up and down the braid.

Here's a trick I use to prevent components (such as headers) from falling out when the board is turned over. I place the board in a vise and add the headers. Using a small piece of cardboard or another circuit board, I place the cardboard on top of

■ **No-Nonsense schematic.**

15 k · R1 · + · 4 Alkaline · Photoresistor · PN2222a · Rainbow leds · etc

the headers, pinch and turn over the "sandwich," then place it on the bench and start soldering (**Figure 5**).

With parts that have uneven heights, I tack one resistor lead to the top of the board, and then turn it over and solder the other lead. Then, I re-solder the tacked lead.

Keep your soldering tip tinned and always add solder to the tip before turning it off. This will help prevent corrosion. Leaving the soldering station on overnight will ruin the tips. I use an automatic one hour turn-off to prevent this.

I have added four mounting holes to the four corners. The two upper mounting holes are connected to the switched positive voltage and the two lower ones are connected to ground. If you need more letters, you can bolt two or more boards together by mating the right mountings of one board to the left mounting holes of another board.

Bolt them together using two 3/8" 6-32 screws with nuts. Since the first board will provide the power, you can
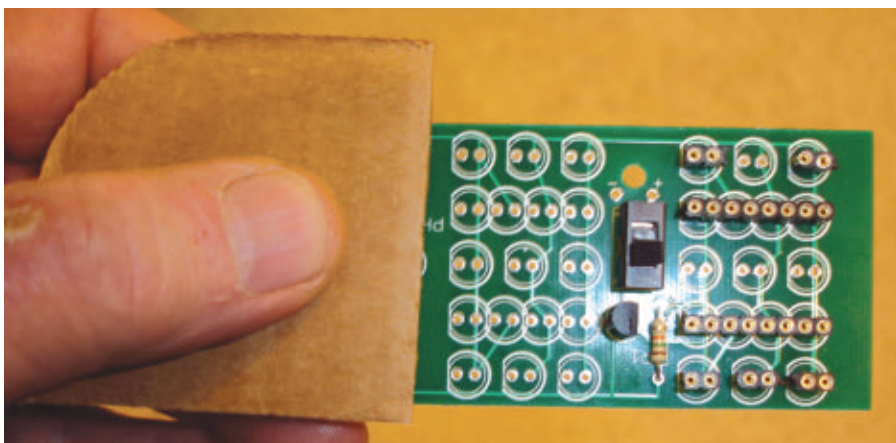
■ **FIGURE 5. Using cardboard.**

eliminate the switch, resistor, transistor, and photocell from the extra board. Make sure you increase the battery capacity because you will be drawing more power.

Well, that wraps up this fun and useful project. Now, folks will be able to read your messages loud and clear, instead of between the lines. **NV**

THE 8x8 RGB LED MATRIX DISPLAY AND THE ARUINO

By Jerry Spring VE6TL

Post comments on this article and find any associated files and/or downloads at **www.nutsvolts.com /index.php?/magazine/article/ february2015_Spring.**

When I bought a couple 8x8 LED displays for about $7.50 each, I had no idea that I would spend the next couple of months trying to figure them out. While there are Arduino shields (for about $20) that will drive this device, it seemed like a lot more fun to figure it out myself. The neat thing about the display is that you get immediate results, and can test out dozens of variations of your sketch in a short amount of time. Plus, the LEDs glow super bright and look really cool.

After initially breadboarding the hardware, I decided to purchase a plastic case, a prototyping printed circuit board (PCB), and a five-pin DIN jack/plug to connect up to the Arduino for future experiments. In addition to learning how the device worked, my ultimate goal was to program it to display my ham radio callsign (VE6TL) and have it scroll from right to left. I even got it to change colors every time through.

The purpose of this article is to challenge your imagination as to what might be possible, and to provide a few ideas as to how to get started.

Getting Started

The first step when tackling any new device is to check the datasheet and try to understand it. While there isn't much available for the particular GTM2088ARGB-24 LED units discussed here, there is just enough information to get it to work.

If you look closely at the datasheet info, you'll notice the red LEDs require only 2.3 VDC to light up, while the green and blue ones require 3.4 VDC. This works out to using 330 ohm resistors for the lines that go to the red LEDs, and 220 ohm resistors for the blue and green ones. If all the resistors were the same value, we would find a mismatch in intensity; this would make it more difficult to blend colors.

Now comes the fun part: figuring out which pins connect to where, and why. Check out the connection diagram also from the datasheet.

The first thing to note is that there are two rows of 16 pins that protrude from the bottom of the device. So, why aren't the pins arranged like the diagram and how can you tell what goes where?

When I purchased my devices, they came with four eight-pin headers that can be used for mounting. I was able to insert these into the breadboard and then mount the display on top of them. However, when I went to mount the display, it became immediately apparent that there were no indications of pin 1 or which way to install it. To solve this, I connected a 220 ohm resistor from the Arduino 5V supply and tested various pins until I found pin 1. You can't damage anything with this approach.

Referring back to the diagram, we note that rows 1-8 (on the left) correspond to the common anodes for each LED. Hence, there are eight pins (17-20, 39-32) that need to be connected to the 220 ohm resistor (that is then connected to 5V). By connecting a ground wire to the pins across the top, we can find out which is pin 1. For example, if we want to light up the green LED in the top left corner, we need to connect the anode at pin 17 and the cathode at pin 28. This leads to the second problem: What is the pin numbering convention? After more checking (with my 5V and ground wires), I discovered the pattern shown in the figure to the right. I don't know if my numbering system follows some sort of convention, but it worked for me.

So far, so good. In order to be able to address all 64 LEDs with three colors each, I would need to be able to turn on and off 32 bits of data simultaneously. This requires four eight-bit shift registers daisy-chained together. At this point, I will refer you to the work of Francis Shanahan[1] who wrote an excellent article on this subject in 2009. Not only does he provide a schematic on how to

Electro-Optical characteristic(Ta=+25℃)

Product Model	Emitted color	Forward voltage (IF=20mA)/VF		Luminous intensity (IF=20mA)VF		Dominant Wavelength (IF=20mA)/λd
		MIN	TPY	MIN	TYP	
GTM2088ARGB-24 (GTM2088ARGB-13)	RED	1.9	2.3	110(100)	120(110)	622～625
	GREEN	3.0	3.4	240(220)	270(240)	522.5–523.55(523.75)
	BLUE	3.0	3.4	70	80	487.5～468.75(468.75～470)

■ **Datasheet listing voltages, intensities, and wavelengths for the LED matrix display.**
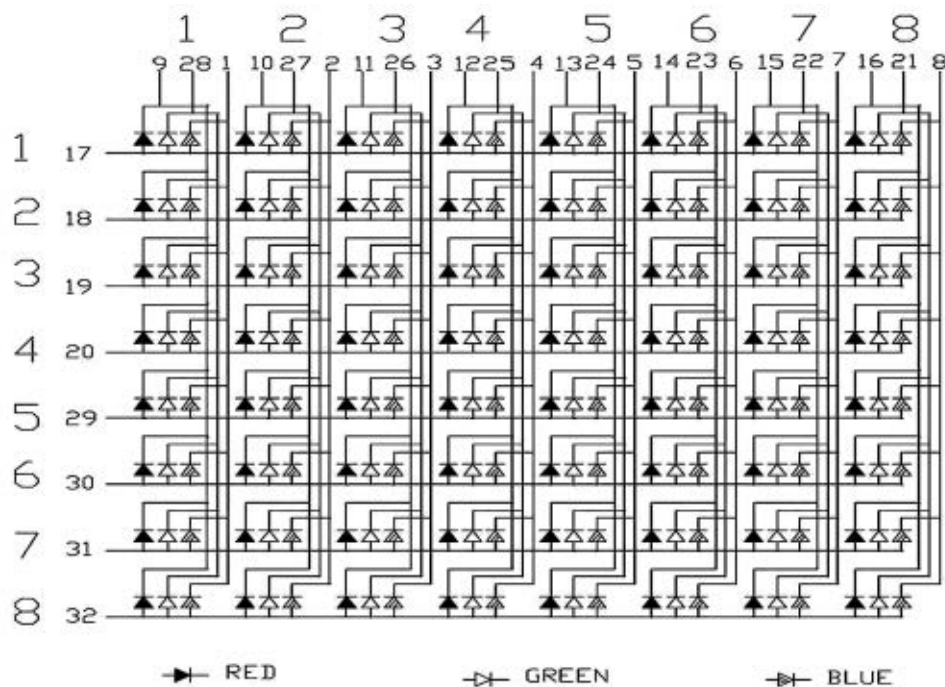
■ **Connection diagram from the datasheet.**

wire everything up, but a link to a lengthy Arduino sketch that makes the display show what it is capable of doing. If you are not familiar with shift registers, there has been plenty written about them, including the *Nuts & Volts* Spin Zone column in the January 2014 issue.

Time to Build

Check out the photo of my breadboard setup. For

■ **Pin numbering of the RGB LED matrix.**
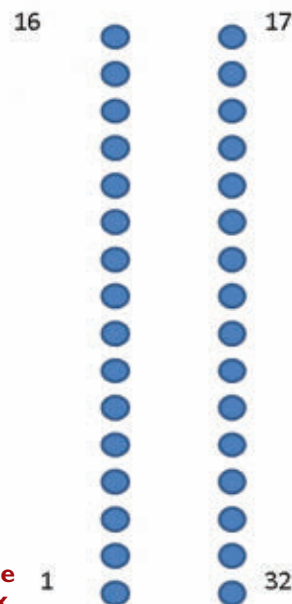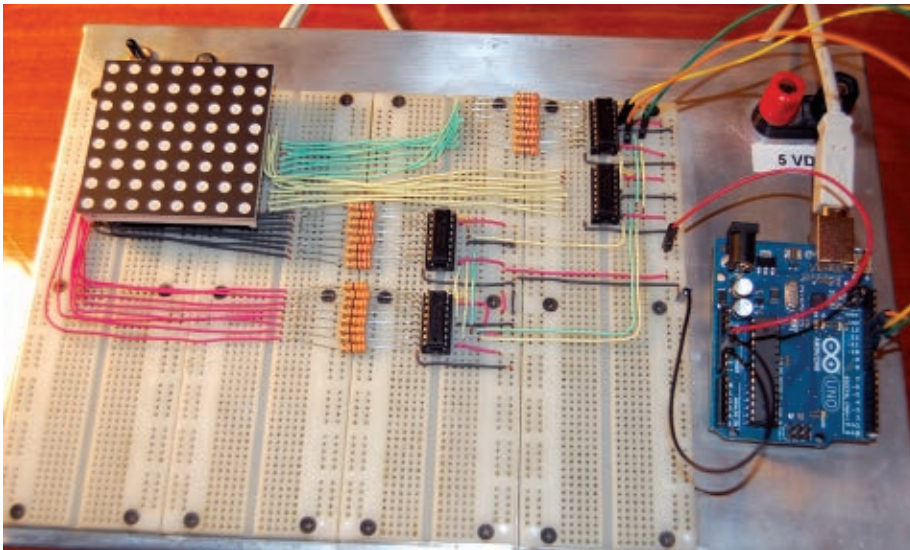
■ Breadboard with Arduino Uno, four 74HC595 eight-bit shift registers, resistors, and display.

■ Finished display case and circuit board that can connect to an Arduino.

find my wiring mistake because it was a simple matter to see where the signal was lost. I would highly recommend obtaining a logic probe if you intend on building circuits like this one.

Testing Things Out

Assuming you've made it this far, it is time to actually connect up the Arduino to the circuit, load a small sketch, and see if everything is working properly. After connecting the five wires from the circuit to the Arduino (see commented lines in the sketch, plus Vcc and GND), it is time to type the sketch into the IDE (integrated development environment), compile it, and load it. The first sketch is designed to simply see if everything is working.

For those who have been working with shift registers and binary data, the sketch should be easy to follow and require little explanation. The key points can be summarized as:

1. Four bytes are necessary to control the matrix display as the 32 bits correspond to the 32 pins on the device.
2. Byte 1 controls the blue LEDs by row, with LSB (least significant bit) = row 1.
3. Bytes 2 and 4 control the red and green LEDs respectively, by row.
4. Byte 3 controls the anodes and the columns.
5. Bytes 1, 2, and 4 turn on rows by setting values to 0 (low).
6. Byte 3 turns off columns by setting values to 0 (low).

For those who build the circuit and get it to work (with the above sketch or their own), they will quickly see that expanding on these ideas can be fun and rewarding. Try to figure out how to address a single LED and move it around the matrix. What about moving columns side to side and rows up and down?

The second sketch moves a single red LED from the bottom to the top, and from left to right. See if you can figure out how it works and then change the pattern to

the wire, I used some old four-conductor telephone cable I had in my junk box, after removing the outer insulation with a utility knife. Over the years, I have come to appreciate that neatness counts when building things like this, as it doesn't take much to get off track. Being subject to Murphy's Law (of course), when power was first applied to the circuit, it naturally failed to work. This is when I finally got to use my "logic probe" — a device I picked up at a local ham radio flea market for a dollar that has been sitting in a box ever since.

The logic probe has two wires that connect via alligator clips to ground and 5 VDC. When you touch the tip of the probe to a point in the circuit, an LED will light up indicating a high or low value. It didn't take long to

1 http://francisshanahan.com/index.php/2009/how-to-build-a-8x8x3-led-matrix-with-pwm-using-an-arduino

■ **Sketch1.**

```
00000 * * * Top of File * * *
00001 /*
00002    First experiment to see if 8X8 Matrix even works.
00003    This flashes all the LEDs alternating rows of red
00004    and white.
00005 */
00006 const int SER = 8; //Serial output to shift register
00007 const int LATCH=9; //Shift registers latch pin
00008 const int CLK=10; //Shift registers clock pin
00009
00010 void setup() {
00011    pinMode(SER,OUTPUT);
00012    pinMode(LATCH,OUTPUT);
00013    pinMode(CLK,OUTPUT);
00014 }
00015 void loop() {
00016    digitalWrite(LATCH,LOW); //Start Sending data
00017    shiftOut(SER,CLK,MSBFIRST,B10101010);
00018    shiftOut(SER,CLK,MSBFIRST,B00000000);
00019    shiftOut(SER,CLK,MSBFIRST,B11111111);
00020    shiftOut(SER,CLK,MSBFIRST,B10101010);
00021    digitalWrite(LATCH,HIGH); //Stop Sending data
00022    delay(100);
00023    digitalWrite(LATCH,LOW); //Start Sending data
00024    shiftOut(SER,CLK,MSBFIRST,B00000000);
00025    shiftOut(SER,CLK,MSBFIRST,B00000000);
00026    shiftOut(SER,CLK,MSBFIRST,B00000000);
00027    shiftOut(SER,CLK,MSBFIRST,B00000000);
00028    digitalWrite(LATCH,HIGH); //Stop Sending data
00029    delay(300);
00030 }
```

something else. How would you change it to a different color or combination of colors?

Final Thoughts

This article represents the tip of the iceberg as to what you can do with this type of display. For example, even though we are using shift registers, we can still use PWM to control the brightness of individual LEDs and achieve the full spectrum of visible colors.

This requires advanced programming techniques and knowledge of the Arduino's internal clocks and interrupts. The good news is that the work has already been done and is available for download.[1]

For someone who is ambitious, he/she could take parts of this sketch and generate an Arduino library similar to those available for LCDs or other output devices, and make it available to the Arduino community.

Finally, when it came to achieving my goal of generating characters and scrolling them across the screen, I was able to do this by: 1) deciding on a seven row x five column font; 2) storing each character in an integer array (16 bits); and 3) using the bit shift operator to scroll the characters across the display. This took a little while to figure out, but the satisfaction in the end was worth it.

After all this work, I decided to build a permanent case and circuit for the LED matrix display. I found that the headers fit so snugly onto the 32 pins of the matrix display that no other means of mounting the board was necessary. A little bit of foam in the bottom kept

■ **Sketch2.**

```
00000 * * * Top of File * * *
00001 /*
00002    Experiment - to program 8X8 Matrix.
00003    Start with red LED bottom left and move to top left
00004    Shift to column two, three, etc. and repeat
00005    Repeat with green and blue
00006 */
00007 const int SER = 8; //Serial output to shift register
00008 const int LATCH=9; //Shift registers latch pin
00009 const int CLK=10; //Shift registers clock pin
00010
00011    int seqa[8] = {1,2,4,8,16,32,64,128};
00012    int seqb[8] ={254,253,251,247,239,223,191,127};
00013 void setup() {
00014    pinMode(SER,OUTPUT);
00015    pinMode(LATCH,OUTPUT);
00016    pinMode(CLK,OUTPUT);
00017 }
00018 void loop() {
00019    //Start with sequencing RED LEDs
00020    for (int i=0;i<8;i++) {
00021       for (int j=0;j<8;j++) {
00022    digitalWrite(LATCH,LOW); //Start Sending data
00023    shiftOut(SER,CLK,MSBFIRST,B11111111); //BLUE
00024    shiftOut(SER,CLK,MSBFIRST,seqb[j]); //RED
00025    shiftOut(SER,CLK,MSBFIRST,seqa[i]); //ROW
00026    shiftOut(SER,CLK,MSBFIRST,B11111111); //GREEN
00027       digitalWrite(LATCH,HIGH); //Stop Sending data
00028       delay(50);
00029    }
00030 . }
```

everything from moving around.

As a future experiment, I've been thinking about making this a stand-alone display by writing the program to a microcontroller chip, mounting it under the matrix display (the headers provide clearance), adding a battery pack, and on/off switch. The possibilities after that are endless.

Good luck with your tinkering and a big thanks to Francis Shanahan for inspiration and a few hints. **NV**

By Craig A. Lindley

MIDI BUDDY

Being a musician, I have known about MIDI (Musical Instrument Digital Interface) for a long time. As a guitar player, I never immersed myself in the technology like most keyboard players did. However, when I read about the SAM-2195 single chip synthesizer, I knew it was time to dive in and learn something new. That is what's fun about electronics and computers; there is always something new to learn. Connecting up hardware devices and writing software to make it all work keeps those little gray cells in our brains active, happy, and alive. If the end result of our experimentation is something useful — like the MIDI Buddy — that is icing on the cake.

MIDI Buddy has been my platform on which to experiment with and learn more about MIDI. This initial foray into MIDI has helped me to understand the physical protocols. Future investigations will focus on using MIDI to produce sound directly using a microprocessor/controller which may, in fact, lead to another article, so stay tuned. For now, our first question to answer is: "What is MIDI, anyway?"

What is MIDI?

MIDI is a technical standard/protocol that allows a wide range of electronic musical instruments, musical equipment, and computers to communicate. MIDI technology was standardized in 1983, and the standard is maintained by the MMA — MIDI Manufacturers Association.

The MIDI specification defines more than a software protocol to control devices. It also defines the connectors and cables which are to be used to convey the protocol.

MIDI messages are made up of eight-bit bytes transmitted serially at 31.25 Kbaud. The first bit of each byte identifies it as a status byte or a data byte, and is followed by seven bits of information. A start bit and a stop bit are added to each byte for framing purposes, so a MIDI byte requires 10 bits for transmission.

A single MIDI connection can carry up to 16 channels of information — each of which could be routed to a different MIDI device.

The MIDI protocol itself doesn't contain sounds, but rather specifies a stream of messages/events that cause MIDI devices to generate sounds. This is why a MIDI file for a song will be much much smaller than, say, an MP3 file of the same tune.

What kind of MIDI events are there, you ask? There are many, but events like NOTE ON when a keyboard key is depressed and NOTE OFF when it is released are simple examples.

MIDI events can also control which instrument or voice plays the note (piano, nylon string guitar, tuba, etc.), how loud or soft the note volume should be, how much reverb should be applied to the note, and so on. MIDI events provide both coarse and fine grained control of musical parameters. This allows the nuances of a musical performance to be captured accurately in a MIDI event stream.

Since MIDI's inception, electronic musical instruments have continued to evolve. MIDI has had to evolve to stay pertinent. The original MIDI specification made connectivity between MIDI devices possible, but didn't define what voicing would be heard for which notes. In 1991, the General MIDI (GM) standard was established which defined a standard sound bank for MIDI devices (which supported GM), allowing a MIDI file created on one MIDI setup to sound similar when played on other setups. In addition, GM established MIDI channel 10 as the percussion channel where specific MIDI note values are mapped to specific percussive sounds. Finally, GM mandated a minimum of 24 note polyphony which would allow complex musical compositions to be adequately expressed.

The evolution of MIDI did not stop with GM. Newer standards like GS, XG, and GM2 were developed to further refine MIDI's fine grain control, and to take advantage of advances in electronic music hardware. These standards are all backward compatible with GM but not necessarily compatible with each other.

So, What is MIDI Buddy?

MIDI Buddy is a stand-alone device consisting of the following hardware:

- A Teensy 3.1 microcontroller
- An SD card interface for up to 2 GB of available storage
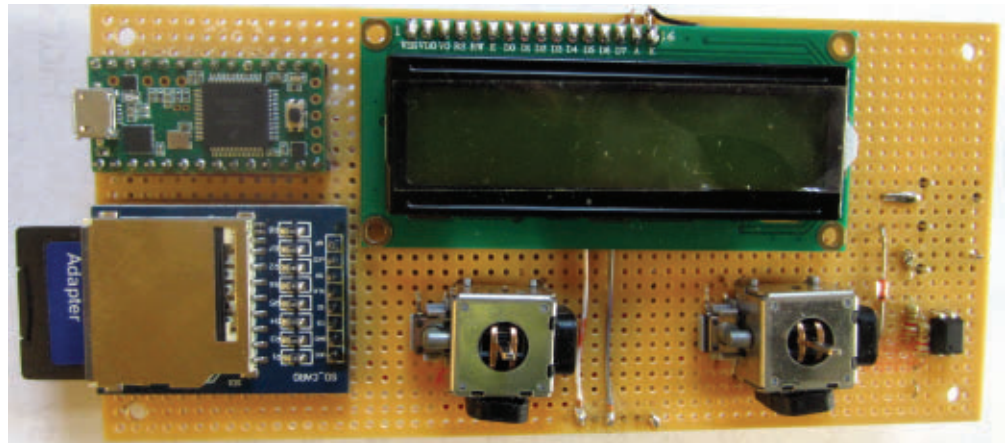
■ **Top view of the MIDI Buddy breadboard. The Teensy 3.1 controller is in the upper left.**

- A two line by 16 character LCD display used for displaying menus and status
- Two joysticks for menu navigation and control
- A standard MIDI input
- A SAM-2195 synthesizer chip
- A stereo output via 1/4" phone jax

MIDI Buddy is powered by a USB power supply plugged into the Teensy controller via a USB cable. The audio output of MIDI Buddy is very high fidelity with rich piano, pads, and string sounds. The sound quality blew me away the first time I heard it.

Visualize MIDI Buddy as a standard MIDI input which feeds MIDI data into a serial input on the Teensy controller. A serial output on the Teensy streams MIDI data to the SAM-2195 synthesizer chip which converts (renders) the MIDI events to sound for output. The SD card interface also feeds MIDI events to the Teensy controller.

Since the Teensy controller is the middle man in all this, it has the ability to:

- Pass the data it receives from the MIDI input through unmodified to the synthesizer chip for rendering into audio.
- Interpret and/or modify the data it received from the MIDI input before passing it along to the synthesizer chip.
- Generate MIDI data programmatically and pass it to the synthesizer chip.
- Read a MIDI file and pass the decoded MIDI events to the synthesizer chip for playback.

With the software I provide at the article link, MIDI Buddy can:

- Function as a complete music synthesizer if a keyboard or other MIDI controller is plugged into its
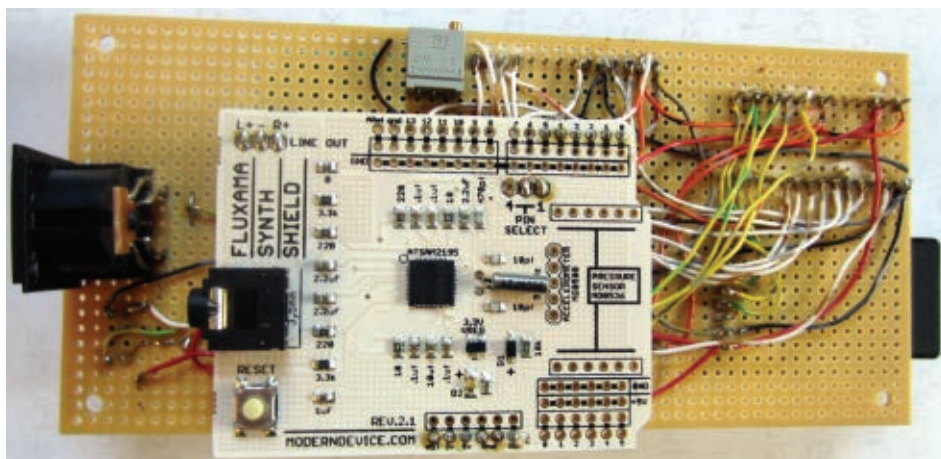
Bottom view of the MIDI Buddy breadboard. The LCD contrast trimmer is the gray component at the top; the MIDI connector is at the left; and the Fluxamasynth shield is in the middle. NOTE: A jumper is soldered to the pin 1 select on the shield and can be seen in this image.

MIDI input. MIDI Buddy can then be played in real time and can sound like a piano, organ, cello, bagpipes, steel drums, or any of the many other instrument voices supported by the synthesizer chip.
• Play any type 0 (single channel) Standard MIDI File (SMF) stored on the SD card. MIDI Buddy could function as a backing band for a solo performer with this functionality.
• Can assign any of the 256 synthesizer voices to any of the 16 MIDI channels.
• Can enhance a musical performance by adding reverb, chorus, and/or delays to the produced sounds.
• Record and playback (up to 1,000 notes) MIDI riffs to/from the SD card. In this respect, MIDI Buddy can be considered a MIDI sequencer/librarian but without note editing capabilities.
• Play four demos which illustrate various aspects of MIDI:
    • **Demo 1** — All voices demo. Plays all available voices using a random algorithmic rhythm.
    • **Demo 2** — Note bending demo. Illustrates how pitch bending in MIDI sounds.
    • **Demo 3** — Cross fading demo. Demonstrates smooth cross fading between two different voices on two MIDI channels.
    • **Demo 4** — Drum pattern demo. Plays a programmatically generated drum pattern.
• Can be used as a platform for MIDI experiments.

Since the Teensy 3.1 controller is running at 96 MHz and has 256K of Flash (program memory) and 64K of RAM (most of which is still unused), a lot more functionality can be added to MIDI Buddy. Almost any MIDI function one could imagine, in fact.

I've mentioned the SAM-2195 synthesizer chip used in the MIDI Buddy several times, so I better describe its capabilities now in a bit more detail. The SAM-2195 is a single chip (about the size of the fingernail on your little finger) with:

• 128 general MIDI instrument sounds and an additional bank of 128 variations built in.
• Four built-in drum kits.
• Supports 64 voice polyphony when not using effects, or 38 voice polyphony with full effects.
• 14 bits of pitch bend range.
• Master volume and per channel volume control.
• Eight types of reverb effects with variable level and variable feedback.
• Eight types of chorus, flange, and delay effects.
• Four-band equalizer.
• Spatial effects.
• Portamento and modulation effects.

When I first read about this chip, I immediately tried to buy some only to find out they are impossible to get in small quantities. What I did find was a company called Modern Device that sells an Arduino shield called the Fluxamasynth containing the SAM-2195 chip for a reasonable price ($35). So, I used this shield in a non-standard way (which I will describe shortly) in MIDI Buddy. If I understand the history correctly, the SAM-2195 chip was originally designed by the US company, Atmel but production has since been taken over by a French company called Dream. One other note: In March 2014, the SAM-2195 was replaced by an even more powerful chip called the SAM-2695. I have yet to find one of these to play with, however.

I should note also that MIDI Buddy doesn't take full advantage of the synthesizer chip's power. I only implemented features I thought would be useful to me (and to you, of course).

Hardware

Figure 1 shows the schematic of the MIDI Buddy. Please refer to it during the following discussion. Since MIDI Buddy would be a one off, it was built using a breadboard and point-to-point wiring as can be seen in the various photos.

As mentioned, the MIDI specification dictates the hardware aspects of MIDI, as well as the protocol aspects.

The MIDI input shown in the upper right corner of the schematic conforms to that specification. Every MIDI device must be electrically isolated from other MIDI devices to prevent ground loops and other problems. In MIDI Buddy, I used an optoisolator for isolation.

As serial MIDI data arrives at the MIDI input, the LED in the optoisolator toggles with the data. The phototransistor in the optoisolator responds to this activity and generates a digital signal which mirrors it. This newly derived digital signal is presented to the Serial 1 receive input on the Teensy 3.1.

A second serial device on the Teensy — Serial 2 — outputs serial MIDI data from the controller to the input of the Fluxamasynth. Software running in the Teensy configures both of these serial interfaces to run at 31250 bps, which is close enough to the MIDI standard rate of 32.25 Kbaud.

The LCD is connected to the Teensy using a simple four-bit interface. Any 16x2 LCD display compatible with the Arduino LiquidCrystal library can be used. A 10K ohm 20 turn trimmer was used for the LCD's contrast adjustment. This trimmer is adjusted to make the LCD readable.

The Fluxamasynth shield was meant to be directly plugged onto an Arduino form factor device. Here, however, the shield is stood off from the bottom of the breadboard with inline connectors; the five connections required between the shield and the breadboard are directly wired. NOTE: A jumper must be installed on the shield to configure the input to the shield. MIDI Buddy requires pin 1 as the input.

The Teensy controller's SPI interface is used to interface to the SD memory card. It seems like every SD breakout board I come across uses a different labeling
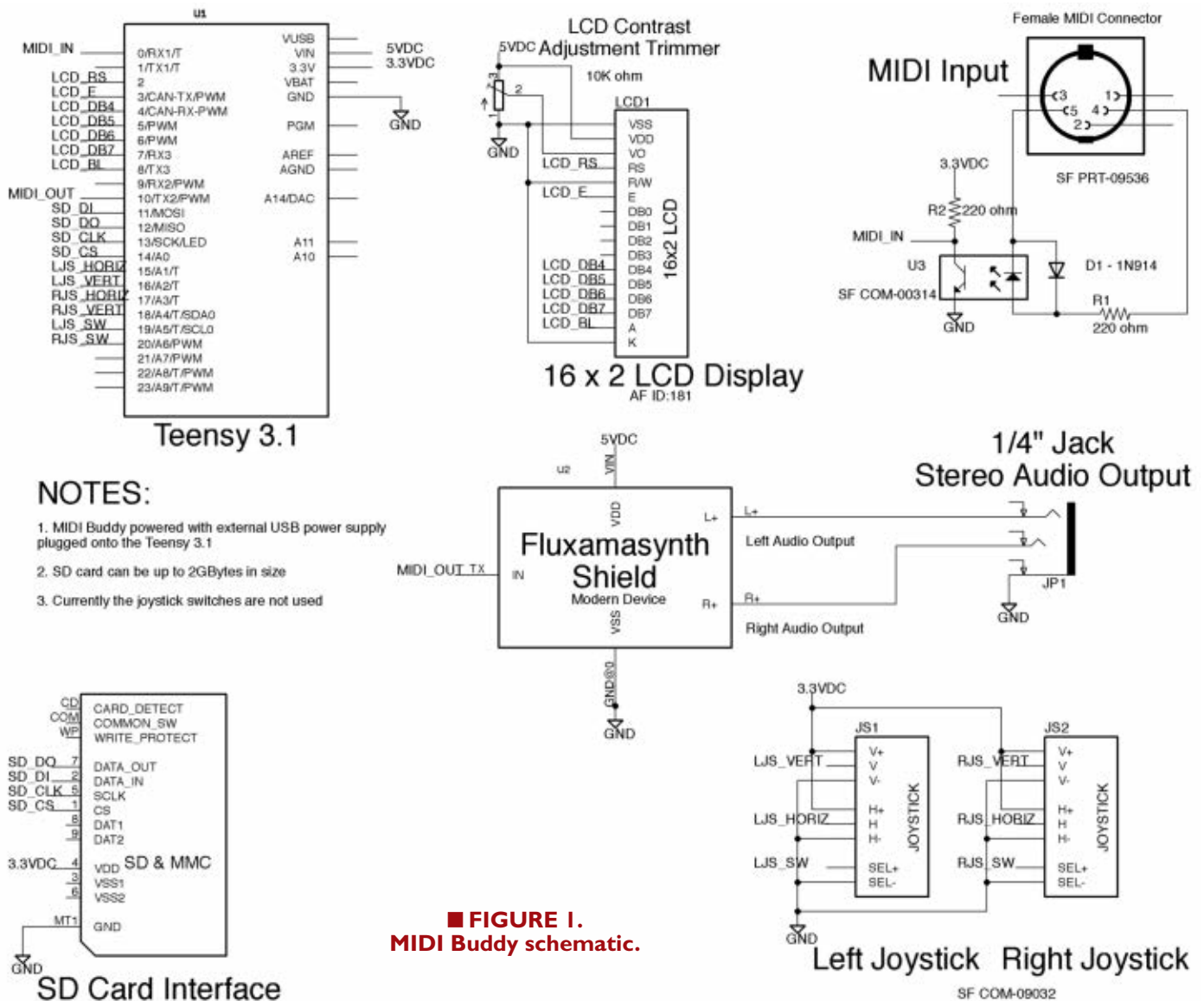
■ FIGURE 1.
MIDI Buddy schematic.

File Name	Function
Demos.h/Demos.cpp	File containing the code for the four MIDI Buddy demos. These demos were ported from the Modern Device library for the SAM-2195 chip.
FileFunctions.h/ FileFunctions.cpp	Caches the names of up to 30 MIDI SM files on the SD card. The limit of 30 was picked out of the air and can easily be changed if access to more files is required.
MIDIBuddy.ino	Main MIDI Buddy sketch.
MIDIInfo.h	Header file with voicing names and effect parameter names. Used in the user interface.
MIDIParser.h/ MIDIParser.cpp	A parser for type 0 Standard MIDI Files.
MIDIPlayer.h/ MIDIPlayer.cpp	Code which takes events generated by the MIDIParser while reading MIDI files, and sends them to the SAM-2195 for rendering.
SAM2195.h/ SAM2195.cpp	An API for controlling the SAM-2195 synthesizer chip.
UISMFunctions.h/ UISMFunctions.cpp	User interface state machines implementing the MIDI Buddy functionality.
Utils.h/Utils.cpp	Various printing and logging functions, in addition to functions for polling and reading the joysticks.

Table 1.

Menu Entry	Function
Play MIDI File	Allows user to browse the available SMFs on the SD card and select one for playback.
Record Riff	Allows user to record a riff played on an attached MIDI device and save it on the SD card.
Play Riff	Allows user to play back a previously recorded riff from the SD card.
Master Volume	Allows user to set the master volume level of the synthesizer.
Drum Set	Allows the user to select which of the four drum sets/drum kits is to be used on MIDI channel 10.
Synth Reset	Resets the SAM-2195 chip to its power-on state.
Chan Volume	Allows the user to set the volume of individual MIDI channels.
Chan Program	Allows the user to set the voicing of individual MIDI channels.
Chan Pan	Allows the user to determine where in the stereo sound field a channels' voice should be heard.
Chan Chorus	Allows the user to configure chorus on a channel by channel basis.
Chan Reverb	Allows the user to configure reverb on a channel by channel basis.
Chan Portamento	Allows the user to configure portamento on a channel by channel basis. Portamento is the glide between notes; from none to very long.
Chan Modulation	Allows the user to configure modulation (tremolo) on a channel by channel basis.
D1: All Voices	Runs Demo 1.
D2: Note Bending	Runs Demo 2.
D3: Cross Fade	Runs Demo 3.
D4: Drum Pattern	Runs Demo 4.

Table 2.

scheme for the signal connections, so please ignore the pin numbers on the schematic and make the connections to the Teensy based on signal names.

You should run the SdFat library example programs to verify you have the SD card connected correctly. Just remember to set the SD card chip select in the example programs to pin 14.

The two joysticks are connected to analog and digital inputs on the Teensy. Each joystick has a side-to-side or horizontal potentiometer, and an up and down or vertical potentiometer. In addition, each joystick has a switch which is activated by pressing down on the joystick. The joystick potentiometers are connected between the 3.3 VDC supply and ground, and the arm of the pot is connected to an analog input. As the joysticks are moved, the voltages at the analog inputs vary, and with that the software can tell which direction the joystick is pointing or if it is active at all. Although connected and configured, the joystick switches are not currently used in the MIDI Buddy software.

Most of the parts used in MIDI Buddy are available from trusted sources including Adafruit, SparkFun, **PJRC.com**, Digi-Key, Jameco, etc. The Fluxamasynth shield I believe is only available from a single source. See the **Resources** section for vendor information.

If you don't mind a longer wait, many of the parts are available on eBay from Chinese sources. Note, that while the prices on eBay are great, the quality can be hit or miss. I bought a game controller with two joysticks at a local Goodwill store for $3.50 and extracted the joysticks for use in MIDI Buddy. Of course, you can buy these new, but why?

I would suggest the Teensy 3.1 controller be purchased directly from Paul Stoffregen of **pjrc.com** who designed it and rigorously supports it. Paul has worked hard making the Teensy usable in the Arduino Integrated Development Environment (IDE) and has ported many of the most popular Arduino libraries to the Teensy. This means you can develop code in the Arduino IDE and download it via USB to the Teensy just as you would if you were using an Arduino. He calls his adaptation of the Teensy into the Arduino environment, Teensyduino. All MIDI Buddy software was developed in the Teensyduino environment.

Part numbers listed on the schematic with an AF prefix are Adafruit part numbers. Parts with the SF prefix are SparkFun part numbers.

Software

As mentioned, the MIDI Buddy software is available as a zip file at the article link. **Table 1** will give you an idea of what each file contains.

The functions provided in the current MIDI Buddy software are described in **Table 2**.

■ **MIDI input and stereo audio output.**

■ **PHOTO 4. Access to Teensy 3.1 USB connector and SD card.**

Most of MIDI Buddy's software is straightforward, but how the user interface (UI) is implemented is somewhat unconventional because it was very important to allow the user to hear changes they make in real time. Keeping the UI responsive while at the same time allowing MIDI data to be received and rendered is a challenge without a multitasking operating system. So, this was accomplished using a series of nested finite state machines which run continually. Anytime the UI is waiting on the user for joystick input, control is returned to the main loop so that MIDI data can flow without interruption. As soon as user input is detected, the state of the UI state machine is changed and the process starts over. Check out the *loop()* function in MIDIBuddy.ino to see how this is done.

MIDI Buddy requires the LiquidCrystal and the SdFat libraries to compile and run. The LiquidCrystal library comes preinstalled with the Arduino IDE, but the SdFat library must be installed. You can find where to get it in **Resources**.

The SD card must be formatted with the FAT16 format before any standard MIDI files can be stored on it. File names must be in 8.3 format; that is a maximum of eight characters in the file name and up to three characters in the file name extension. Remember only single channel type 0 MIDI files (the most common type) can be played by MIDI Buddy.

Operation and Navigation

I pondered many different user interface schemes for MIDI Buddy, but finally settled on using two joysticks largely because of the cool factor. The left joystick is used to select a function for MIDI Buddy to perform. Clicking

Component	Description	Source
Teensy 3.1 Microcontroller	Microcontroller with 256 KB Flash and 64 KB RAM	
USB Pwr/Prog Cable	USB five-pin Micro-B cable	PJRC; SparkFun; Adafruit
USB Power Adapter	5 VDC @ 1A	SparkFun; Adafruit
LCD Display	16x2 line LCD display compatible with Arduino LiquidCrystal library	SparkFun; Adafruit
10K ohm Trimmer	20 turn trimmer for LCD contrast adjustment	RadioShack
Female MIDI Connector		SparkFun PRT-09536
220 ohm Resistor	1/4 watt resistors; R1 and R2	SparkFun; Adafruit; RadioShack
Optoisolator		SparkFun COM-00314
Diode	1N914 or 1N4001; D1	RadioShack
Fluxamasynth Shield	SAM-2195 synthesizer chip shield	Modern Device
1/4" Stereo Jack	Audio output connector	RadioShack
SD Card Interface	SD card breakout board	SparkFun; Adafruit
2 GByte SD Card	SD memory card	SparkFun; Adafruit; RadioShack
Joysticks	Joystick; JS1 and JS2	Recycled or SparkFun COM-09032
Build Material	Perf board, wire, solder	
Packaging Material	1/4" MDF, acrylic sheet, spacers, screws, nuts	

PARTS LIST

the left joystick up or down scrolls through the possible functions (enumerated earlier) which are shown on the LCD. Clicking the left joystick to the right selects the function. Clicking the left joystick left does nothing.
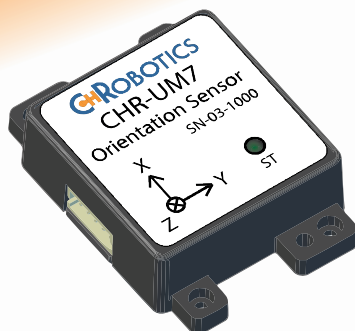
Once a function is selected, the right joystick becomes operational. You use the right joystick to increment and decrement values, and to select additional parameters needing configuration. If you click the right joystick to the left, you immediately backup one menu level.

As a simple example, click the left joystick up or down until Master Volume is shown on the LCD. Click it right to select this function. With the right joystick, move up and down to set the desired volume which will be shown on the LCD. You will be able to hear the volume changes in real time if you have a MIDI source plugged in and are listening to MIDI Buddy's output.

Once you arrive at the volume level you want, click the right joystick to the right; the volume will be set and control will return to the top level MIDI Buddy menu.

Learning about MIDI has been fun and enlightening, and it has also shown that old dogs can still learn new tricks. Have fun building a MIDI Buddy of your own and jam on! **NV**

Resources

The following sources provide more information about this project and its background.

Information about getting started with the Arduino can be found at **http://arduino.cc/en/Guide/HomePage**. The Arduino IDE can be downloaded from **http://arduino.cc/en/main/software**. Note version 1.0.5 or newer of the IDE is required for use with the MIDI Buddy.

For information about and/or purchase of the Teensy 3.1 microcontroller board, go to **www.pjrc.com**. There is also a forum for information about the Teensy line of microcontrollers at **http://forum.pjrc.com/forum.php**. Questions regarding the Teensy should be directed there.

The Teensyduino (version 1.18 or newer) software development add-on to the Arduino IDE is available at **www.pjrc.com/teensy/teensyduino.html**, as well as instructions for installing it.

The Fluxamasynth shield containing the SAM-2195 synthesizer chip is available from Modern Device. Info is at **moderndevice.com/product/fluxamasynth-shield/**.

A datasheet for the SAM2195 chip is available at **http://serdaco.com/files/SAM2195.pdf**

The SdFat library is available at **https://github.com/greiman/SdFat**.

For background information about digital audio, see my book *Digital Audio with Java*, published by John Wiley & Sons.

Other microcontroller based projects of mine can be found at **www.craigandheather.net/celepage.html**.

A Simple Ferro Electric Memory
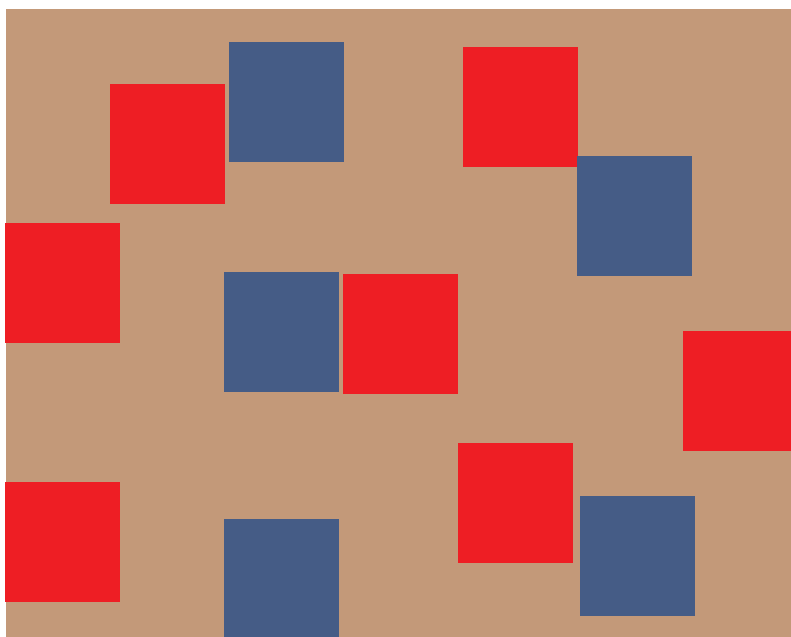
I'm going to describe how to construct your very own two-bit non-volatile memory using a seemingly magical material. I hope that you find this subject so fascinating that you demand more information ...

By Joe T. Evans, Jr.
radiant@ferrodevices.com

Post comments on this article and find any associated files and/or downloads at **www.nutsvolts.com/index.php?/magazine/article/february2015_Evans**.

Introduction to Magic

Magic, of course, is technology we don't understand. Once understood, it ceases to be magic; it becomes a cell phone or a computer or an MRI scanner. A peek inside any of these machines always puts me in a state of awe.

They all run on a magic crystal which has been purified beyond belief, and formed into a single lattice so perfect that it makes a jewel-quality diamond look like a dirty window. As potters have done for thousands of years, we paint minerals on this crystal, fire it in the space-age versions of kilns, and cover it in a glaze to protect its innards from the poisons in our sweat. This magic crystal is silicon.

Using Einstein's entreaty that geometry is everything and calculating the required dimensions using the quantum physics that Einstein helped create but never believed, we package up tiny silicon crystals of amazing variety and configurations, connect them together, and perform feats today that would have been considered magic in 1954 (when I was born). Imagine my disbelief a decade ago when I called home to New Mexico from a car on a highway in Korea. This magic is now the everyday mundane.

Fear not, more magic is on the way! Lots more!

Physics is the fundamental basis of all science. (Just ask any physicist! He or she will gleefully assure you of this perceived truth.) However, physics has different disciplines. The one in which we are interested is solid state physics: the study of materials. From this discipline springs our knowledge of silicon and how to bend it to our will. Solid state physicists — who date to a time long before the Lycurgus Cup of 400 A.D. (look it up!) — know thousands of materials with electrical or mechanical properties that appear at first to be magical. In this article, I will introduce you to just one: lead zirconate titanate, or PZT.

While silicon circuits think logically, PZT feels, detects, moves, and remembers. Connecting PZT devices to silicon circuits is like connecting a body to a brain. PZT and a whole host of related materials called ferroelectrics are poised to greatly expand the usefulness of silicon circuits — even to the point of allowing them to move of their own volition.

Radiant Technologies has quietly worked on this technology since 1988 to make it useful not just for large manufacturers, but to the individual as well. We now manufacture discrete versions of ferroelectric devices that you can insert into circuits for experimentation. The concept of using highly non-linear materials in such a

Figure 1:
Single-bit non-volatile ferroelectric memory attached to a microprocessor and the associated UP/DOWN ferroelectric hysteresis loops.
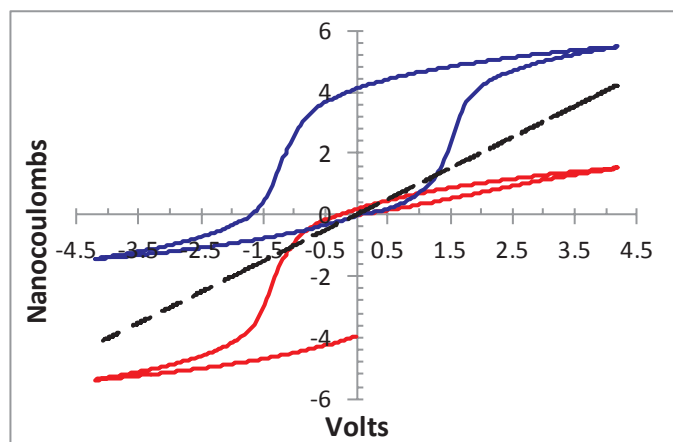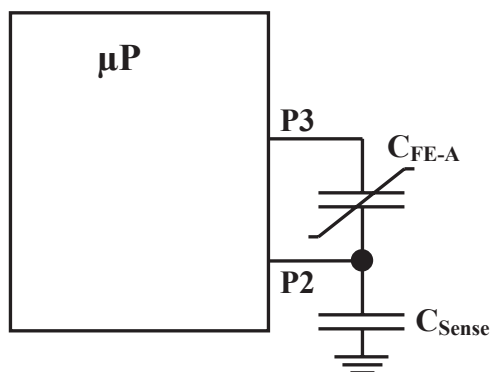
manner is so new that there are hardly any examples of useful circuits. The power of these materials is mesmerizing, and it is my hope that you will experiment with them and kick out new ideas along the way.

The first discrete ferroelectric devices I will discuss are capacitors, but they are to traditional capacitors as transistors are to diodes. You have a lot to learn and there are no text books! Therefore, we will start with the simplest of applications: non-volatile memory.

The project discussed here creates a discrete two-bit ferroelectric memory that operates in the same manner as the Ferroelectric RAM (FRAM) embedded in products now made by Texas Instruments, Cypress, Fujitsu, and Rohm. First, I will explain the fundamental properties of ferroelectric capacitors, i.e., where does the memory come from? I will then show how to connect them to a microprocessor to store and read non-volatile data.

This project should challenge the very way you think about electronic components. Where a diode or a capacitor or a transistor is a cold-hearted mechanistic object that never varies from the equations physicists say it should follow, a ferroelectric capacitor is born with memory and always has memory. A ferroelectric capacitor is never empty. The circuit here simply orients that ferroelectric memory and later asks the capacitor to tell us what it knows. Let's get started!

Ferroelectric Theory

Unlike EEPROM or flash solid state memories that shoot electrons into insulators, the crystal lattice of a ferroelectric capacitor is naturally polarized. Each unit cell of the lattice is a small electric dipole that points towards one or the other electrode.

A dipole is defined as the spatial separation of "+" and "-" electrical charges. Dipoles come into being in traditional capacitors when a voltage is applied, but they go away again at zero volts. In ferroelectrics, the dipoles

are always present. The unit cell is about 4 Ångstroms on a side, so even a ferroelectric capacitor only one micrometer on a side (one hundredth the width of a human hair) has billions of internal natural dipoles within its volume. Each dipole can be rotated by an external voltage causing distortion in the lattice that remains when voltage is removed.

Think of the lattice distortion as an over-center lock that holds all of the dipoles pointing in the forced direction. A ferroelectric capacitor need only have all of its dipoles point in the same direction to form a detectable memory state in that capacitor. Applying the opposite voltage to the capacitor switches the dipoles to the opposite direction. The alignment of the dipoles in a single direction results in a net electric field inside the capacitor.

That field attracts and holds extra charges onto the capacitor plates to prevent any dipole-generated electric field from escaping the capacitor volume. It is from the movement of that extra charge to/from the plates as the dipoles change direction that we can determine the memory state of the capacitor.

Many readers will remember building a crystal radio that used a quartz crystal to detect and de-modulate an AM radio signal. The quartz crystal could accomplish this magical feat because it too has a charged lattice like a ferroelectric crystal. The difference between the two is that the ferroelectric material allows its internal dipoles to be rotated in different directions while quartz does not. This property of quartz makes it an electret material, as opposed to a ferroelectric material like PZT.

Non-volatile memory using ferroelectricity is charge based, not voltage based. Its state is destructively read by counting electrons coming from the capacitor when it is forced into a known direction. The value of the state (1 or 0) is related to how many electrons come out. Did the capacitor switch the dipole direction or not? The material itself is not damaged by these operations nor is its ability to be re-written affected.

The charge coming from a ferroelectric capacitor is most easily converted to a voltage for detection by a linear capacitor in series with the ferroelectric capacitor. This arrangement is known as the Sawyer-Tower circuit. A non-volatile ferroelectric memory bit may be created by connecting a Sawyer-Tower circuit to two I/O pins of a microprocessor to take advantage of the unique property of microprocessor pins, whereby they can act as active outputs or as high impedance level-sensing inputs. Microprocessors with comparator or ADC (analog-to-digital converter) pins will allow even more robust operation of the Sawyer-Tower circuit.

The circuit in **Figure 1** shows a Sawyer-Tower circuit connected to two I/O pins of a microprocessor. The plot in **Figure 1** shows two single-cycle hysteresis loops of the ferroelectric capacitor starting from opposite dipole orientations. Note that all hysteresis plots shown in this article are measurements of actual devices. They are not hand drawn loops or simulations. The horizontal axis is the voltage applied across the ferroelectric capacitor. The vertical axis is the count of the charge that comes from the capacitor during actuation.

All measurements start with the Sawyer-Tower sense capacitor initialized to zero volts, so the memory state of

the capacitor determines vertical positioning on the Y axis. If the dipole direction does not switch, only a small amount of charge comes out, and it goes almost back to zero like a standard capacitor when the voltage recedes. If the dipoles do switch, a lot of charge comes out with the positive voltage, but not all of that charge can go back in when the voltage returns to zero. In order for all of the switched charge to go back in, the dipoles must switch back on their own, but they don't! Since the ferroelectric capacitor ends up with all dipoles pointing positive after the application of the positive voltage independent of the capacitor's original state, full dipole switching always takes place during the negative voltage application portion of either loop.

As a reference, there is a third loop in the plot of **Figure 1**. The straight gray line is the hysteresis loop of a very linear 1 nF capacitor. The formula for a capacitor is Q=CV. For a linear capacitor, the capacitance term in the formula is constant, so this becomes the formula for a line centered on the origin with slope equal to the capacitance. The comparison of the linear capacitor hysteresis with the ferroelectric hysteresis clearly highlights the magic of a ferroelectric capacitor. The plot also gives the perception of the total charge that comies out of this

Figure 2: Write and read operations, and the associated signals on P2.
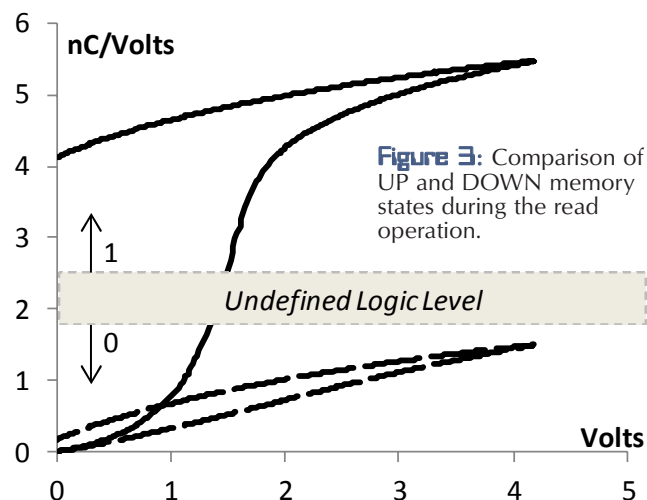
Figure 3: Comparison of UP and DOWN memory states during the read operation.

particular ferroelectric capacitor. Some would argue that a linear capacitor does not have hysteresis. I maintain that zero is a number, so a linear capacitor has a hysteresis loop!

From the diagram, it can be seen that if the ferroelectric capacitor has all of its internal dipoles oriented DOWN prior to the positive half of the loop, switching occurs in a single pass. However, if the remanent polarization is pointed UP before the loop

begins, only a very small non-switching loop occurs in the first half of the cycle.

The difference between the two half loops in the positive quadrant of the plot is the difference in voltage that will develop across the sense capacitor of the Sawyer-Tower loop during a read operation. The value of the sense capacitor of the Sawyer-Tower circuit should be selected so that the UP memory state will not generate enough voltage across P2 to be read as a high logic state, but the DOWN memory state will. Once the read operation wipes out the stored datum, it must be re-written afterwards.

The four parts of **Figure 2** show the Write UP, Read UP, Write DOWN, and Read DOWN operations and their associated pin assignments.

Figure 3 combines the Read UP and Read DOWN results into a single plot. Csense should be sized so that the UP signal is less than the logic 0 threshold for the microprocessor input, while the DOWN signal must be greater than the logic 1 threshold.

Discrete Ferroelectric Capacitor

Radiant Technologies manufactures several types of
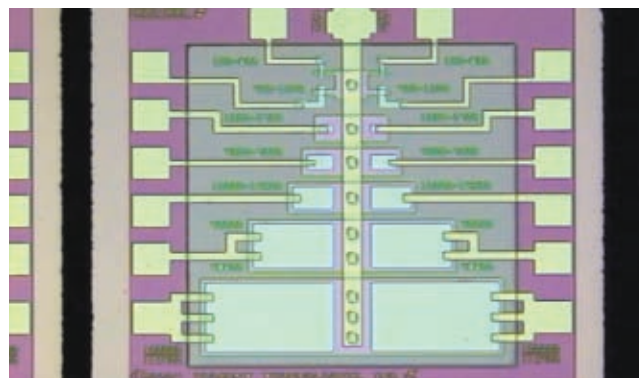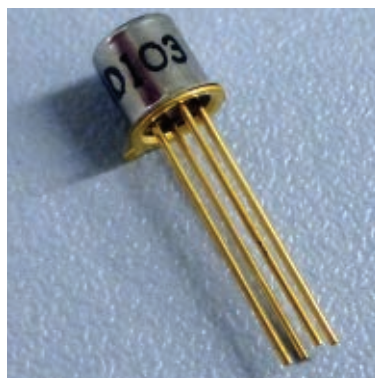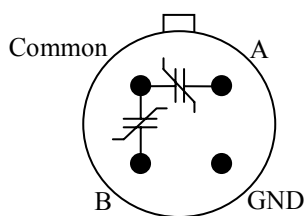
Figure 4: TO-18 package, top-view pin assignments, and a photograph of the die.

Common
A
B
GND

discrete capacitors, and then packages matched pairs in old-style TO-18 four-lead transistor packages (refer to **Figure 4**). The capacitor that should be used with microprocessors is the Type AD, which saturates above 2.2 volts and can operate at 4.2 volts for about a year at 85°C.

The capacitor used in this article has 10,000 square microns of area (Type AD103), making it 2,500 to 40,000 times larger than the ferroelectric capacitors used in the FRAM memory arrays in TI, Fujitsu, Cypress, and Rohm products.

Capacitor properties do not change with a larger area, but the amount of charge produced by the capacitor tracks linearly with the area.

If the ferroelectric capacitor doubles in area, then the Sawyer-Tower sense capacitor must increase its capacitance proportionally in order to maintain the same sense voltages.

For detailed information on how to select the sense capacitor and mathematically determine circuit performance, go to **www.ferromems.com** and click on the link for "Theory - Ferroelectric Memory."

The AD103 capacitor works most easily in logic mode with a 1 nF to 2 nF sense capacitor.

Microprocessors

The Arduino Uno microprocessor (**Arduino.cc**) shown in **Figure 5** with the Sawyer-Tower memory circuit can be configured to work with the Type AD ferroelectric capacitor to form an external non-volatile memory bit.

The amp® microprocessor by Parallax (**www.parallax.com**) in **Figure 6** is also an excellent choice.

Both are available from a variety of sources, are inexpensive, have five volt I/O pins, and are easy to program from any PC by USB connection.

Ferroelectric capacitors made with PZT are extremely fast. The Type AD capacitors can be written to either state in less than 100 nanoseconds.

Thus, the most difficult problem associated with using them in non-volatile memory is to prevent stray voltages from being applied inadvertently. The slightest blip you cannot even detect can fully switch a capacitor that should not be switched. This is especially true during power changes.

You cannot use a microprocessor that does not default all I/O pins to input mode during power-up. At the same time, the I/O pins connected to the Sawyer-Tower circuit must be set as inputs prior to turning off power. It is good practice to set these pins as inputs whenever the ferroelectric capacitor is not being accessed.

When the pins are to be activated as outputs, they should first be set to zero volts out before being changed from input to output. They should both return to zero volts after a read or write operation before changing them back to input mode.
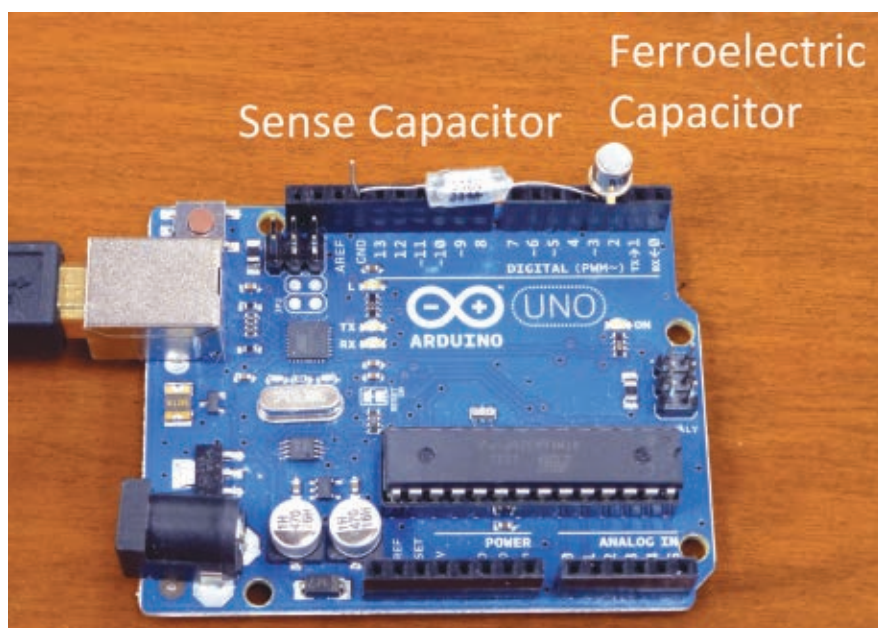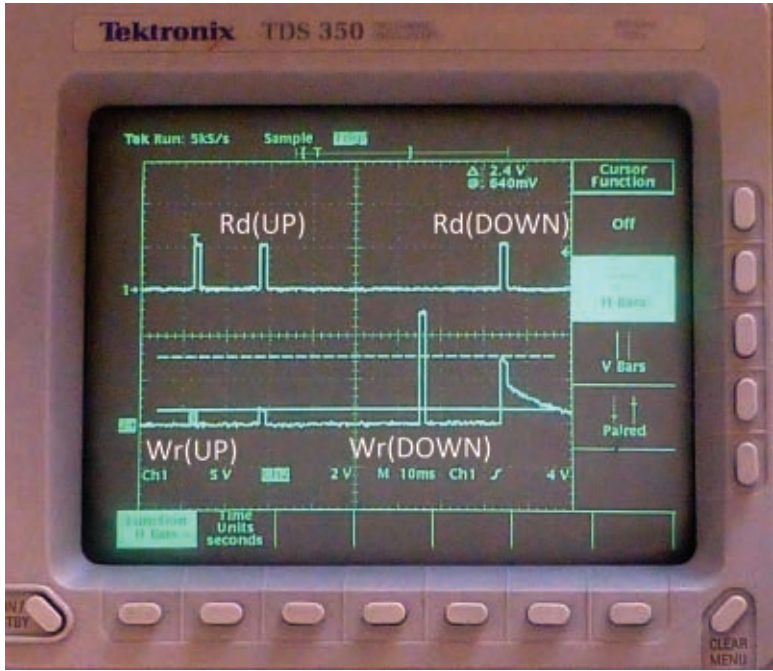

Figure 5: Arduino Uno with Sawyer-Tower circuit.

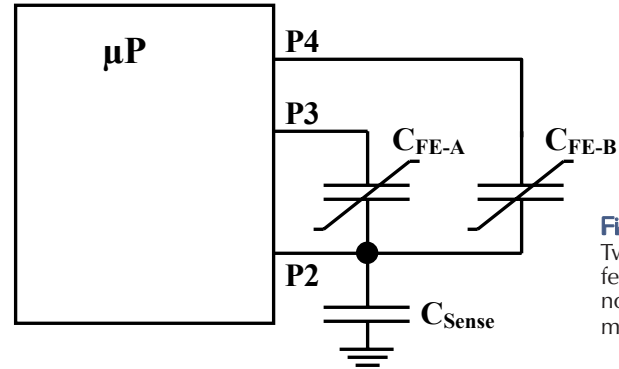| Operation	Instruction	P3	P2
Write UP	1	0	0
	2	1	0
	3	0	0
Write DOWN			
	1	0	0
	2	0	1
	3	0	0
Read			
	1	0	*Input*
	2	1	*Input*
	3	1	*Read*
	4	0	0
	5	Re-write	

Table 1: Generic I/O commands to operate a simple ferroelectric memory.
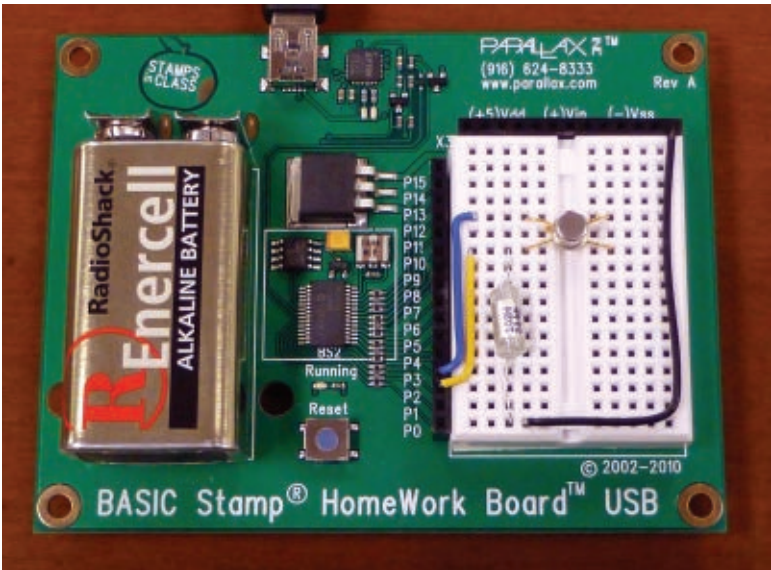
Figure 6: BASIC Stamp with P3/P2 connected to a Sawyer-Tower circuit.

Figure 7: Oscilloscope trace of P3 and P2 for a Write/Read UP/DOWN sequence.

Figure 8: Two-bit ferroelectric non-volatile memory.

Thus, every operation involves the following order of steps:

1. Write "0" to both pins while they are set as inputs.
2. Change pins to outputs.
3. Execute the read or write operation.
4. Return to "0" on both pins as the last step of any operation.
5. Change both pins to inputs.

Table 1 uses the pin definitions shown in **Figure 1**. Note that the P3/P2 pin assignments in **Table 1** match the circuit layout on the BASIC Stamp in **Figure 6**. The pin assignments are reversed on the Arduino in **Figure 5**.

Figure 7 shows an oscilloscope image for the P3 and P2 signals for the write/read operations of the two logic states on the Stamp. Sample code files for both microprocessor types can be found at the article link.

Read operations are destructive just like in DRAM operations. The FRAM memory read procedure always leaves the UP condition after the operation is completed. If a DOWN state was previously stored in the bit, it should be re-written after the read operation is completed.

Note that speed is not a consideration in programming the write and read pulses. PZT capacitors physically switch faster than the command execution speed of either processor. The only speed limitation arises if the ferroelectric capacitor is so large that it loads the current

delivery capacity of the I/O pin. For example, suppose the I/O pin can deliver 20 milliamps which is defined as 20 millicoulombs per second.

Figure 3 shows that the 10,000 square micron AD103 capacitor requires slightly less than six nanocoulombs to saturate. A little arithmetic indicates that a 20 mA I/O pin should be able to saturate the AD103 in approximately 300 nanoseconds! Neither the Stamp nor the Arduino can execute a digital pulse on an I/O pin that fast.

of the memory properties of the ferroelectric capacitor.

This is necessary in order to move to the next step: a non-volatile latch using a ferroelectric capacitor that needs no microprocessor to execute reads and writes.

While a few bits of non-volatile memory operating from the I/Os of a microprocessor is not very exciting, an autonomous non-volatile latch is just the opposite! **NV**

Two Bits Instead of One

The Type AD capacitor package contains two identical capacitors. By connecting the second capacitor to its own I/O pin, a two-bit non-volatile memory is created (refer to **Figure 8**).

The same subroutines as above are used to access either capacitor in the package using P3 or P4 to select the capacitor address. P2 operation remains the same. If accessing P3, P4 must be set as an input to prevent voltage application across the P4 capacitor from the P2 node. If accessing P4, P3 must be set as an input.

Conclusion

Once the two-bit ferroelectric memory has been constructed, you can perform a variety of experiments or use the memory during operation of the microprocessor. For example, what four states do you want your garage door to remember?

Since the information is embedded in lattice distortions of the ferroelectric material, it is very robust. You can remove the two-bit package from the circuit after writing data, and carry it in your pocket without losing the written information. Plug it back in the next day to see if the data is still there.

The functionality of a discrete non-volatile memory using ferroelectric capacitors attached to a microprocessor is easy to understand. It is an excellent vehicle for developing a solid understanding

Designing Antenna Systems

By John Belliveau KA1WRH

Post comments on this article and find any associated files and/or downloads at
www.nutsvolts.com/index.php?/magazine/article/february2015_Belliveau.

The Radio Antenna System

An antenna is more than just copper wire or aluminum tubing. It is part of a system. Typically, an antenna is the business end of a radio system. It starts with the radio receiver or transmitter, and connects through a length of suitable cable. In some cases, it needs a matching (**1**) device to transfer radio energy efficiently, which produces an ample signal and sends it out to the world.

Here are some thoughts on assembling a radio antenna system with a focus on reducing the known variants that are usually either overlooked or that can creep in when you're not looking. This discussion is focused on simple and available components. No math here — just some helpful observations from my tests over the years as mentioned.

Of course, this is not the only method of setting up an antenna system by a long shot, as there are many good resources (**2**, **3**) out there. Be sure to check the References in the sidebar.

The Parts of the System

The Antenna Itself

Antennas have some interesting characteristics. One of the important ones is the value of the antenna's impedance at its feed point. For practical purposes, designers try for an impedance value around 50 ohms.

One of the main reasons is efficient power transfer. Since the impedance at the radio's output connector is typically 50 ohms and readily-available coax cable is 50 ohms, we start there. Though other cable values are available — ranging from 75 ohms to 600 ohms — these are generally used in situations not covered here. These higher values can also require matching devices to transform the value to 50 ohms.

The value of impedance at the antenna feed point depends on many variables — not all of which can easily be controlled. This makes antenna system implementation a mystery to most — or at best, a trial and error fest. So, let's look at those things we can control and that follow a few simple considerations.

One thing I've noticed is that when radio operators use antenna tuners — whether manual or automatic — they are placed near the radio which is convenient from an operational standpoint. This works to a point, but is not the most efficient approach. A better

Antennas — whether you buy one or make one — are part of a system. In this article, I'll discuss the radio antenna and its role in a radio system. The information expressed here is based on my observation and tests over several years as an amateur radio operator, avid antenna experimenter, and inventor.

For simplicity, my focus here will be based on newer radios with 50 ohm outputs and readily-available 50 ohm coax cable. Many other antenna/coax combinations exist and will be mentioned briefly.

practice would be to put the antenna tuner at the antenna.

The reasoning is simple, but not always practical. If you have a radio with a 50 ohm output connected to a 50 ohm cable, you have an efficient power transfer or — as some call it — a good match (low SWR [4]). With the antenna tuner at the unknown antenna feed point, it can do its job matching the 50 ohm cable to the antenna feed point impedance for optimum power transfer. Ideally, with an antenna of 50 ohms, the tuner isn't necessary unless you are operating the antenna on multiple frequencies.
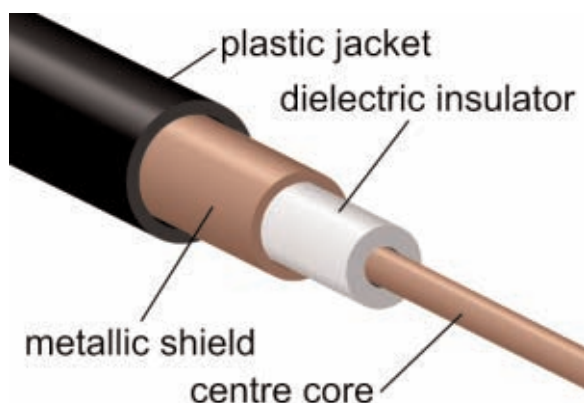
Coaxial cable.

Cable Connection

Of course, 50 ohm coax cable isn't the only option to feed an antenna system. However, it is just a conveniently available one. The main concern here is the quality of the product. Coax tends to have power losses associated with it. This is especially true at higher frequencies and with longer runs, and especially with an antenna impedance mismatch (high SWR).

So, the obvious suggestion here is to buy the best quality coax you can afford, keeping the length as short as possible. That means going to the larger diameter types that generally have lower losses. You lose some flexibility with the larger diameters, so plan for routing through and around things with this in mind.

As stated previously, power loss in the coax can be significant for long runs when the impedance mismatch (SWR) at the antenna is large. If your antenna system is going to be used for VHF or UHF frequencies, this is especially true. Since antenna tuners for these higher frequencies are not readily available, the quality of the coax is very important and, again, the shorter the length of the cable, the better.

Connectors

Personally, I prefer to use N type connectors over the more common UHF connector for any antenna above 50 MHz because I find the N connectors more robust and a bit more weatherproof. They have a consistent match to 50 ohms well into the UHF frequencies. Be especially carefully with inexpensive UHF type connectors, as I have found they can create problems all by themselves.

I also recommend waterproofing any connection that is outside. I make sure to tighten the connector firmly and then wrap it first with a self-sealing tape, then wrap it again with waterproofing tape that can handle UV exposure. If you have not done this and notice a

fluctuation in signal strength or reflected power (SWR), check your connections. Before I used this double tape method, I was able to drain water from inside UHF connectors on several occasions.

Variations of the Theme

Matching Devices Other than Antenna Tuners

Baluns (**5**) — short for "*bal*anced to *un*balanced" — typically are devices used to connect a two element antenna like a dipole or its twin lead feed line (the balanced part of the antenna system) to an unbalanced coax cable. The term balanced antenna here can also mean an antenna that is electrically in balance with the surrounding environment — not necessarily an antenna with two equal length elements.

A balun can have an impedance matching ratio of 1:1, 4:1, 6:1, or even 9:1. The matching ratio can be a current or a voltage mode, with the current mode being preferred because of its broader performance characteristics.

For example, a dipole (**5**) — two equal length elements — may have an impedance at its center feed point of 72 ohms. Here, a 1:1 balun would provide a good match to 50 ohm coax. However, this isn't the only variant here. There is a third signal we need to consider.

Things that Creep In (Besides Water)

Coax has a center lead conductor surrounded by another conductive shield. The transmitter signal operates between the two inner conductive surfaces of the coax. The external part of the shield is exposed to the radiated signal. Since RF signals prefer to ride close to the top surface of a conductor — called the skin effect — the signal can return to the transmitter through this outer path causing RF feedback at the radio. A simple and effective way to reduce this effect called common mode current (**6**) is to coil a few turns of the coax just below the feed point; this is called a choke balun. Ferrite cores are a more effective way of reducing this common mode effect, but depending on the frequency of operation and the power

John Belliveau KA1WRH was first introduced to an antenna around the age of 12. His father helped Belliveau get an old shortwave radio running, and showed him that with a 30 foot piece of wire attached he could hear signals from all over the world. Belliveau set out to understand how that all worked then, and continues to explore all the possibilities.

References

1. John Huggins KX4O — *Myth Busting of Rogue RF Currents*;
**https://docs.google.com/presentation/d/1uJ_INHlJEXUFeDrEiBg
RcpQRKZ3TNSx6ahN_lWdX4ms/edit?pli=1#slide=id.p14**.
2. The American Radio Relay League, *Antenna Handbook,*
22nd Edition.
3. www.dxengineering.com/techarticles.
4. http://en.wikipedia.org/wiki/Standing_wave_ratio.
**5. www.k7mem.com/Electronic_Notebook/antennas
/dipole.htm**l.
**6. www.dg7ybn.de/Symmetrising/Symmetrising.htm#Classic_
Balun_for_Folded_Dipoles**.
7. www.AlphaNodeHub.com.

level, care must be taken in their selection.

The Counterpoise Gift

If we have a reasonably well-designed antenna — one that can utilize a ground plane such as an elevated vertical with radials — it is possible to put a counterpoise plane approximately 1/4 wavelength below the radials. This will also reduce the effect of the RF continuing down the outside of the coax.

J-pole antennas and long wires are very susceptible to this common mode effect and will be greatly enhanced by some method of choking the effects of RF on the outside of the feed line.

The Antenna Environment

The antenna support — whether a mast or strung from a tree — will require a plan, or at least you should think about it before you start.

Antenna masts — usually a metal tube attached to your house, chimney, or (if you're lucky enough) a tower — all become part of the antenna system.

Other metal objects near the antenna are part of the system also. These include rain gutters, metal door frames, cars in the yard, and (in my case) barbeque grills.

Obviously, the closer these objects are to the antenna (1/8 wavelength or less), the more effect they will have.

For HF, this isn't usually too much of a problem, although a 1/2 to 3/4 wavelength distance or more away from the antenna is desirable.

For VHF and UHF, care must be taken to keep the antenna as far away as possible from metallic objects — like BBQ grills.

Grounding Your Mast and Antenna

Grounding a mast for static discharge is a very good idea on any antenna system. The caveat here is, does the

now grounded mast become a tuned part of the antenna system, thus changing the resonance of the antenna and changing the SWR?

When mounting a VHF antenna on a universal mounting hub (7), I assemble the antenna, hub, and radials to a tripod mount. It turns out the upper part of the telescoping tripod was isolated from the lower part. The upper mast becomes a tunable sleeve, and as I move the mast up or down, it changes the resonant point and the SWR of the antenna.

Conclusion

The antenna is part of a radio system and needs to be considered in the whole approach. The antenna, cable, matching device, and environment are all part of this system.

For a simple approach to assessing an antenna as part of a system, I would first look at the frequency or frequencies of operation, the environment, the connection, and the antenna. The variants can be minimized, and hopefully have you operating successfully with minimal frustration. **NV**

By Thomas Henry
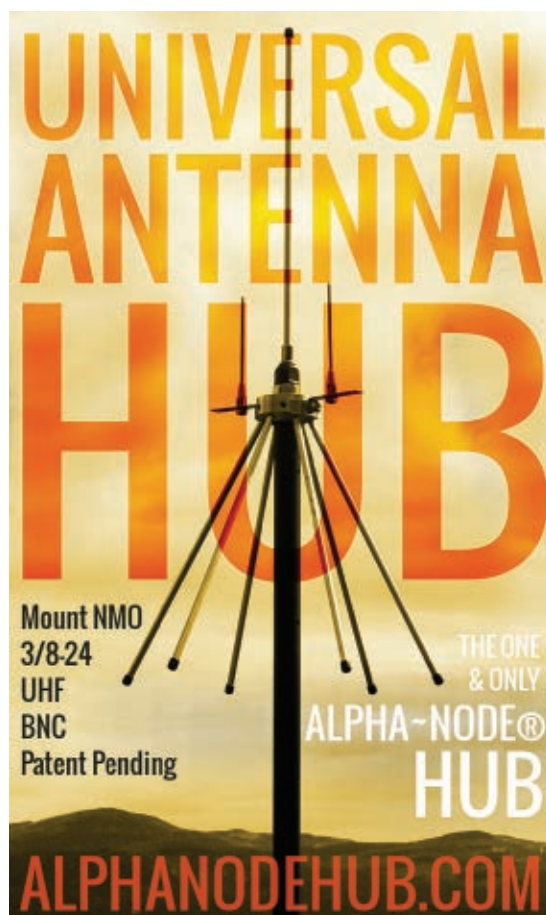
All About PIC Analog Comparators

Analog comparators, though not particularly sexy, must surely rate as one of the more useful building blocks for a wide variety of circuits. They can be used to generate pulse width modulation, monitor under- or over-voltages, shape up ugly signals into nice crisp pulse waves, act as logical inverters, help with interfacing mechanical switches, and more. A comparator has two analog inputs and a lone output, which can be considered digital since it can assume only two possible states. If the voltage on what's referred to as the non-inverting input exceeds that on the inverting input, then the output snaps high; otherwise, it remains low.

It's easy to forget that most PIC microcontrollers sport one or more uncommitted analog comparators. For example, after my article "An Easy Two-Wire LCD" appeared in *Nuts & Volts* (February 2014, pp. 30-36), all of a sudden it hit me (a slap on the forehead moment) that the transistor inverter required there could be completely eliminated — including its two resistors — and replaced with an internal comparator within the driving PIC chip. Just like that, the parts count shrinks by three components, and at no extra cost.

Apart from forgetting to make good use of these comparators, there's that business of the datasheet. On the one hand, I'm delighted Microchip (the manufacturer of PICs) has given us such comprehensive materials to work with. Yet, who hasn't dragged their feet when first approaching this huge daunting manual?

That's where this article comes in. Besides reminding you of the utility of comparators, it will reorganize the concepts of the datasheets into something more approachable. We'll get the big picture in mind first, and only after that will we proceed to the details.

Along the way, seven experiments give you a chance to really nail everything down; these are the very tests I performed to confirm what was going on. With just a handful of common components at your side and a couple sessions at the breadboard, you'll be all set to design your own PIC circuits exploiting analog comparators.

The Big View

As mentioned, just about all of the PIC chips contain at least one analog comparator available for use in various modes or configurations. To keep things specific here, let's

consider the ever-popular PIC16F88 — one of the most commonly employed microcontrollers. Do keep in mind that similar arrangements are apt to apply to whichever specific PIC chip you're keen on.

There are so many options available that it's easy for your eyes to just glaze over and make you want to walk away from the mess. In my 30 years of teaching, I've always found branching tree diagrams useful for organizing multiple options. Somehow, visualizing the choices graphically makes them all less daunting, and the details won't swamp you.

Let's begin by referring to **Figure 1** which illustrates the relationships among the eight possible modes available for the two comparators within the PIC16F88. Each mode is indicated by a three-bit binary number (more on this later).

FIGURE 1.

All eight of these modes are available in the PIC16F88 microcontroller. Simpler chips may be missing some of the modes, or only sport a single comparator.

Only mode 110 makes the comparator outputs available to the pins. In all other cases, you read the current state from the CMCON register.

Only mode 010 uses the internal voltage reference. In all other cases, an external reference is required.

The figure really does tell most of the story quite well, so I won't waste many words here describing what you can see for yourself. Let me just point out a few of the more salient particulars.

Starting at the top of the diagram, you'll see that the comparator modes split into two main categories: Single and Dual. In Single mode, only comparator 2 is enabled. Comparator 1 is completely disconnected from the PIC (via internal multiplexing), freeing up its associated pins for other uses.

On the other hand, with any of the Dual modes, both comparators are available. The Dual modes can be divided into two major categories: Independent and Common Reference. As the name implies, in an Independent mode the two comparators are completely divorced from one another and can be treated as separate circuits. A Common Reference mode has the two non-inverting inputs ganged together. Typically, this frees up a pin for other applications in the PIC.
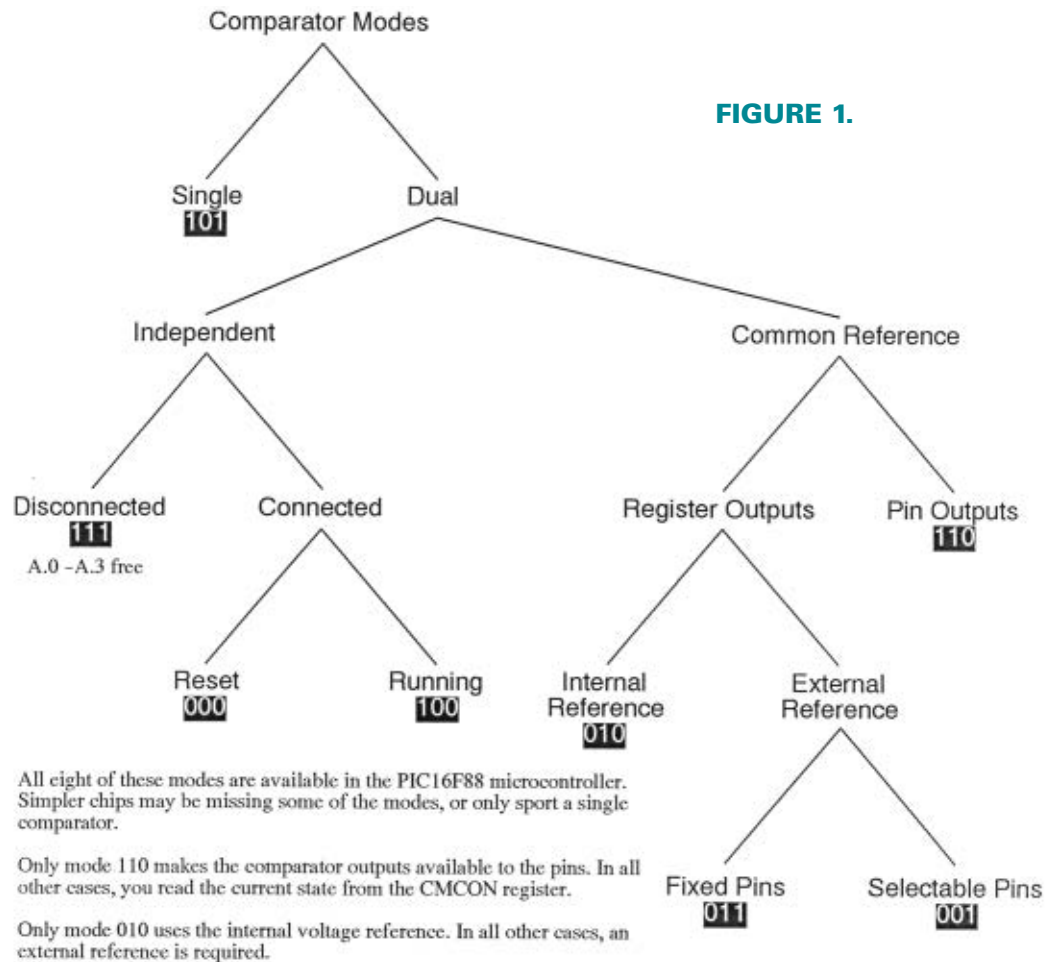
Continuing our tour of the tree, Independent mode splits in two again — either Disconnected or Connected. If the comparators are disconnected, they are indeed disabled altogether, and their pins (port pins A.0 through A.3) are liberated for other general-purpose uses. This is the default situation at power-up, which is why many of us forget about the comparators!

If the comparators are, in fact, connected, then the internal multiplexer brings the inputs out to the chip pins. As **Figure 1** indicates, there are two Connected modes: Reset and Running. When the mode is configured to Reset, the comparators are still hooked up, but their outputs are forced to be zero regardless of the state of the inputs. Otherwise, in Running mode they are free to operate independently of whatever else the microcontroller is up to.

That takes care of all the Independent modes. Move over to the Common Reference modes mentioned earlier. You get two main choices here: Register Outputs or Pin Outputs. In the former, the outputs of the comparators are

CMCON

C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

CM1 and CM2 are the comparator register outputs.

C1INV and C2INV flip the sense of the outputs.

CM0 through CM2 determine the mode.

CIS determines which pins are connected in modes 001 and 010 only.
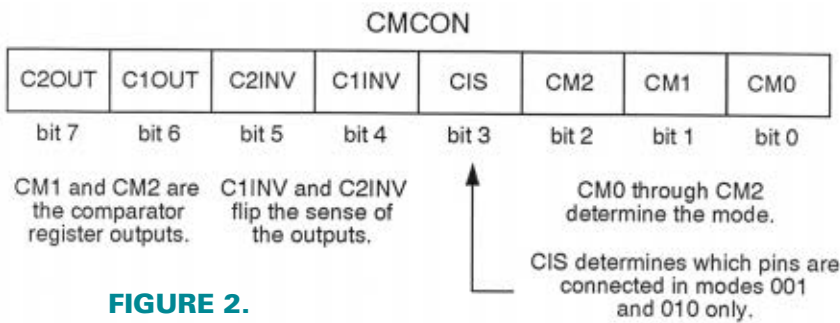
FIGURE 2.

only accessible from within an internal register (CMCON) to be described in a moment. In the latter, it is also possible to route the outputs to physical pins on the PIC16F88, in which case the comparators behave like any outboard unit you may have used in the past.

Now, the Register Output modes bifurcate, giving the ability to sense either an Internal Reference or an External Reference. In either case, the reference voltage will be applied to the non-inverting inputs of the comparators.

Finally, when using an External Reference, you get a choice of either the default pins that Microchip has designated, or are offered some flexibility on which ones actually connect up to the inverting inputs. That's thanks to the magic of multiplexing. This would make it possible to choose one of two voltages to monitor under software control; for example, sort of like an SPDT switch.

How to Select a Mode

Setting each of these eight modes is a snap by means of the CMCON register. The acronym stands for Comparator Module Control, of course. This is illustrated in **Figure 2**. Take a look at bits 6 and 7 first. These are the comparator output bits alluded to previously. They are always in operation, diligently following whatever the two comparators are up to. Clearly, they are read-only in nature.

Bits 0 through 2 form the three-bit mode number mentioned above. Did you notice these codes in **Figure 1**? Just pop the desired number in here and away you go.

Bit 3 is only required by modes 001 and 010, and lets you select which pins actually connect to the PIC comparators. You'll get a chance to see it used in the experiments.

Finally, bits 4 and 5 provide a nice little piece of flexibility. These guys change the sense of the comparators under software control. For example, if bit 4 is cleared to zero, then comparator 1 behaves as you'd normally expect. Make it a one, and then it becomes an inverting comparator.

This is sort of like exchanging the inverting and non-inverting inputs, and is particularly useful in keeping the firmware less convoluted to the eye.

What About Interrupts?

Designing circuits and software to handle an interrupt generated by the comparators is remarkably straightforward. Here are the few things you need to know.

There are three levels of interrupt enable flags (sometimes called masks in other processors). At the deepest level, you need to set the flag CMIE (which stands for Comparator Module Interrupt Enable) when you do indeed want interrupt action.

You'll find this in the register PIE2 — an acronym for Peripheral Interrupt Enable 2. Obviously, the comparators are considered to be peripherals.

One step up from this is the PEIE bit, or the Peripheral Interrupt Enable. It too must be set to get things cooking. Look for it in the register named INTCON — a symbol for Interrupt Control.

Finally, at the highest level, you'll need to set the flag GIE, which is an abbreviation for Global Interrupt Enable. This one is also found within INTCON.

Putting it all together, if you really do want the comparators to generate interrupts, then set CMIE, PEIE, and GIE. If any one of these is clear, then the interrupts are masked.

One final thing. If you look inside register PIR2 (Peripheral Interrupt Register 2), you'll notice the bit CMIF, which stands for Comparator Module Interrupt Flag. This bit is constantly watching the comparators and is set any
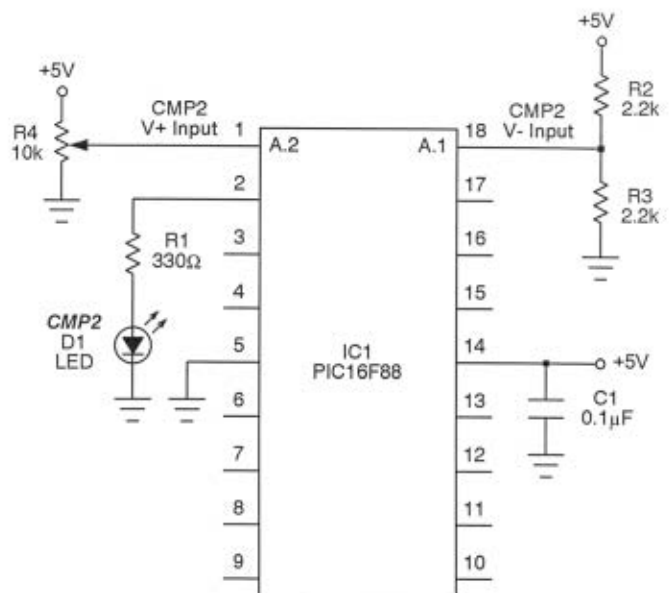
FIGURE 3. This exploits mode 101. Comparator 1 is disabled, with A.0 and A.3 freed up for either analog or digital uses.

time a state changes — regardless if the interrupts are masked or not. When you enter a comparator interrupt routine, be sure to clear this flag before returning to the main program.

Don't be disheartened if it sounds a bit messy! As you'll see in Experiment #7, it's remarkably easy to make this stuff all work.

Experiment #1

Let's stop all this yammering and head straight to the workbench! Your first step is to visit the article link and get the firmware source code. The programs have been created with the excellent and free open source compiler, Great Cow Basic which I've mentioned often in previous articles.

The syntax is exceedingly similar to PICBASIC PRO should that be your weapon of choice. Heck, even porting it over to C shouldn't be all that onerous, since I've annotated and commented the source code like crazy.

Check out **Figure 3**. In Experiment #1, we get to see mode 101 in operation. Recall that only comparator 2 is utilized, with the other being disabled. A reference voltage of +2.5V (courtesy of divider R2/R3) is applied to the inverting input (at pin 18) as the reference.

Now, while monitoring the wiper of potentiometer R4 with a multimeter, watch what happens as you increase the voltage on the non-inverting input at pin 1. Once it exceeds +2.5V, LED D1 snaps to attention.

Observe that since comparator 1 is out of the picture, port pins A.0 and A.3 are freed up for any other use you have in mind. This is a simple experiment, but a great way to get your feet wet.

Experiment #2

Now, turn to **Figure 4** which shows the layout for the next experiment. We're in Dual mode now, with both comparators doing their thing. In particular, you'll get to see them running normally or have their outputs forced to reset.

A +1V reference is applied to comparator 1 at pin 17, while +2V is put onto comparator 2 at pin 18. Again, while monitoring the wiper of either R6 or R7 with a multimeter, observe how LEDs D1 and D2 respond once the threshold voltages are reached.

Closing switch S1 puts the mode to 000, which forces both comparators to reset — regardless of what the potentiometers are up to.

FIGURE 4.

When S1 is open, the comparators are running in mode 100. When closed, the mode changes to 000, and the comparator outputs are forced to zero (in CMCON).

FIGURE 5.

Remove R5 and R6 for Experiment #5

Experiment #3: Mode 011
Experiment #4: Mode 001
Experiment #5: Mode 010

Experiments #3, #4, and #5

The next three experiments use the schematic in **Figure 5**. In all cases, the comparators are instructed to behave as inverters, but now we'll let the non-inverting inputs be the reference. For example, LEDs D1 and D3 will light oppositely from one another, since comparator 1 is acting as an inverter now. Pressing pushbutton switch S1 will toggle these back and forth. S2 behaves similarly with the second comparator. If you want more details, they're in the source code. In Experiment #3, mode 011 is used which gives a fixed pin arrangement and an external reference (provided by divider R5/R6) applied to the ganged reference inputs of the comparators.

Mode 001 is utilized in Experiment #4. This implies that reference pins 17 and 18 could be assigned elsewhere if desired. Again, multiplexing makes this all possible.

For Experiment #5, completely remove voltage divider resistors R5 and R6 because an internal reference of +2.5V will be handily created by the PIC. Refer to the source code to see how easy it is to generate the desired voltage. If it isn't clear, we're using mode 010 now.

Experiment #6

You'll find the circuit diagram for this experiment in **Figure 6**. At last, we're using the pin output feature provided by mode 110. In particular, the pulse width modulation unit within the PIC16F88 applies a varying rectangular pulse to the green element within bicolor LED

D1. Simultaneously, that signal is pumped through comparator 1 acting as a logical inverter. Its output is then applied to the red element of D1.

This implies, for example, that when the green element is seeing a 75% duty cycle signal, the red element is getting a 25% signal. The program sweeps these back and forth, giving a pleasing morph of green to orange to red and back again, over and over.

Experiment #7

This is a neat little demonstration, and really shows off the power of Great Cow Basic simultaneously. **Figure 7** shows the schematic. In this case, we're using interrupts and coercing the PIC to behave as a sort of retriggerable one-shot. Press pushbutton S1 briefly, and LED D1 will light for one second. Press and hold the switch, and the LED simply stays lit until the button is released.

Utilizing comparator interrupts is particularly easy, as the source code readily shows. Give it a quick read-over and see for yourself.

Some Lessons Learned

When I first timidly approached the PIC datasheet, I fell into the trap of misinterpreting a number of concepts concerning the analog comparators. Then, there was the swirl of too many details thrown at me all at once. After working the experiments, I found that everything behaved very simply after all, and reliably to boot. Let me relate just a few of the things I picked up along the way; maybe they'll save you some headaches.

• The comparators are disabled by default at power-up.
• There is no need to diddle at all with ANSEL, the analog select register. Setting the mode in CMCON takes care of configuring the input pins as analog.
• Do set the data direction in TRISA for the input pins.
• Any pins not actively used by the comparators are available for **any** other purpose.
• The comparator inputs always read as zero (not that you'd want to read them directly anyway).
• The output impedance of whatever is driving the comparator inputs should be less than 10K.
• Altering a mode midstream in a program may fire an interrupt.
• Any change in either comparator (low to high or high to low) triggers an interrupt.

When I first looked into the PIC comparators, I think I probably scared myself based on what I was seeing in the

This exploits mode 110.
A.3 is an actual pin/physical output for Comparator 1.
Comparator 2 is ignored.

FIGURE 6.

This exploits mode 101.
Comparator 1 is disabled.
Interrupts are used.
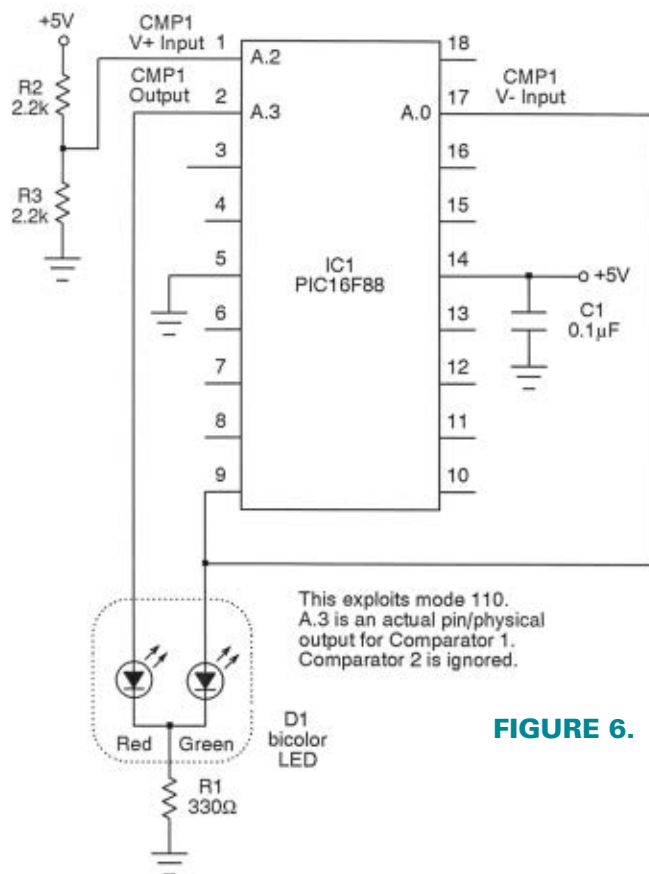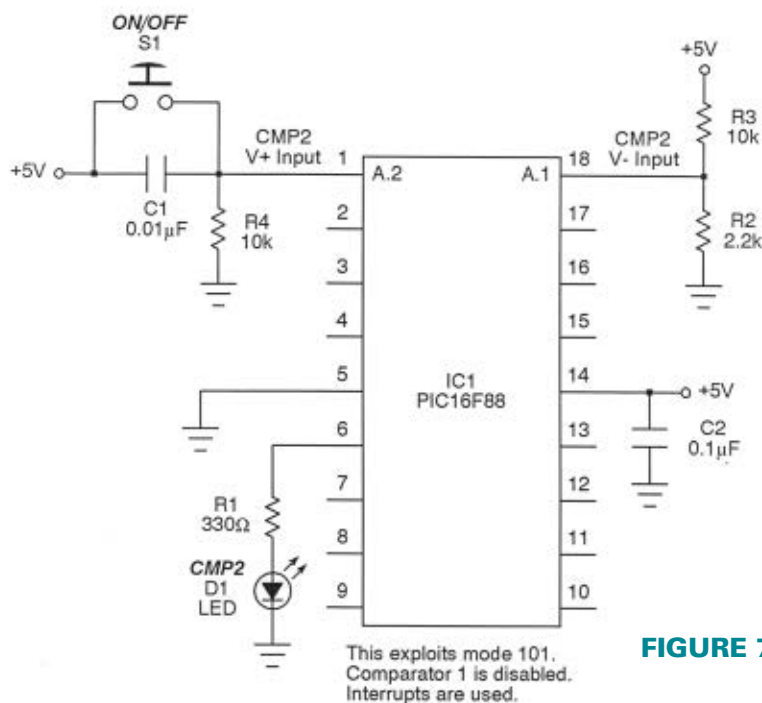
FIGURE 7.

datasheet. However, I hope you agree now that, in fact, they're a breeze to work with.

Run the experiments and prove it to yourself! **NV**
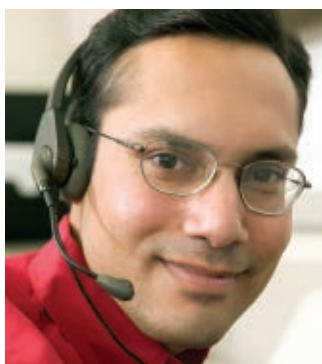
Getting the Most from Tech Support
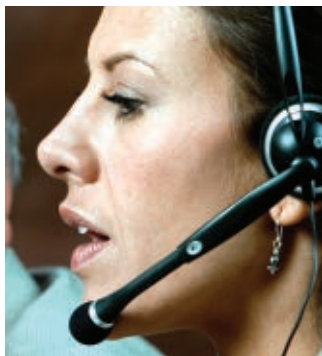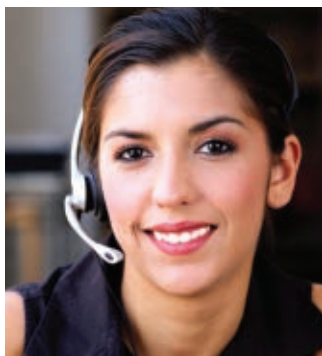
With technology being such an integral part of our lives these days, the chances are that at some point you will need technical support for something. It might be for your Internet service, cell phone, car, or computer. Your tech support experience will probably be based on one of two things: how well the company is at supporting their product, and how well you are able to communicate the problem to them and understand their response.

As an engineering technician, a big part of my job is providing technical support for my company's products. As a consumer and end-user of technology, I also have found myself in need of tech support. Let me share with you some of my suggestions for how to get the most from tech support, should (when) you need it.

Different Types of Support

There are five main types/methods of support that I am familiar with: by phone, email, on forums, the Internet, and service calls. I'll be discussing the first three here.

Common Issues

Regardless of which type of support you're trying to get, here are some common things that apply in almost every situation.

Be prepared to answer some questions or to try some things out. You might get asked seemingly simple questions like, "Have you tried turning the device off and on again?" or "Have you verified the battery or voltage level?" These may seem ridiculous, but you would be surprised to find out that more often than not it is the simple things that are overlooked.

When asked about battery voltage, I have been told many times that the batteries are "new." However, a quick check with a multimeter often shows low voltage — perhaps due to a bad cell. Just

By Chris Savage

because the power LED is on does not mean the circuit has sufficient voltage to operate.

LEDs can light up with a few volts, but often the main circuit has a minimum functional voltage, so you can't be sure until you check it with a meter. Don't check the individual batteries, but instead check the voltage where it enters the circuit. This can often reveal other issues such as damaged power cables or battery holders.

If your device was working and suddenly stopped, ask yourself what changed. Perhaps a connection came loose or a lead on your breadboard is touching something else. If you can't solve the issue, break things down into the simplest terms. Remove everything that has been added to take additional possibilities out of the equation. If the device is battery powered, it may simply be low batteries. Batteries should always be tested under load.

It is a good practice to go into a tech support situation well armed. You do this by trying the seemingly obvious things before you solicit help. Take a voltage measurement just to be sure. Make a note of the voltage in case you're asked for it. Try "rebooting" the device. If it connects to a PC, try re-installing the software or drivers if the device is not recognized. If you have access to another PC, try the device on it to rule out that the PC you're using doesn't have the problem.

Often with driver and Windows updates, this can be an issue with devices that rely on or are programmed via a PC. The more things you have tried and the more knowledge you have, the less time it will take to work out a solution.

If you are in a situation where you have access to more than one of a product, try swapping out the device to help narrow down the cause. I deal with this all the time with sensors, for example. A customer may say that one of their IR sensors is not working. At this point, it is easy to swap the sensor in question with a working one to see if the problem follows the sensor. It may have been a related component actually causing issues.

Be sure of the part number of the product you're asking for support on. I have often been given a product name and started troubleshooting only to get to a point where I realize it is not the product stated. A lot of times products have similar names, so this happens often. A product number or model number helps to clarify the correct product.

Phone Support

When it comes to phone support, you usually get an automated system, a human on the other end, or both. I have dealt with some companies that have completely automated phone support systems with what is essentially a voice interface knowledge base. You navigate a series of menus or, in some cases, you ask questions or state your problem, then are presented with pre-programmed responses. These are often frustrating to deal with, but there's little you can do about it, unfortunately.

Even when there is a human, the common thing these days is to try and direct the customer to the website to solve the problem. When you do get a human for technical support, there are some important things that can help you.

You may have spent considerable time trying to get your product working and now you are frustrated. Don't take your frustration out on the support person. Whether it is due to the problem you are calling about or the wait to talk to them, the support person isn't at fault. They are there to try and help, and a frustrated customer can be difficult to assist — especially if their focus is on time lost or other things that don't contribute to solving the issue.

Cursing is something you want to avoid when on a tech support call. You may be comfortable with it in your personal life and even the tech support person may be comfortable with it in their personal life. However, not everyone is comfortable with someone they don't know using a lot of curse words to describe their issue. It can come off sounding like excessive frustration toward the tech support person which also doesn't help get the issue solved.

Try to limit your information to just the details of the issue. The tech support person is trying to get the information necessary to determine how to best solve your problem. Describing your use of the product or previous experiences often cloud the details the support person is trying to acquire. In some cases, this is what contributes to the wait time on the phone.

Give your first name when you call. Not only does this help with communication but it can also help if you have to call back with a follow-up question since the tech support person may remember you.

I log as much information about each call as I can such as date, time, name, and what the issue was, as well as any information I might need if the customer will be calling back. So, giving your first name can sometimes help avoid having to repeat what you are calling back about.

Sometimes it also helps to give the state you live in or location. This tells the tech support person what time zone you're in, so if they're returning a call you won't get it during dinner — or worse — before you would normally wake up for the day. Technical support typically crosses time zones.

When getting phone support, try to have your product set up and ready to go. Calling for support on a product while driving, at work, or in any situation where you can't try the suggestions the tech support person gives you can be frustrating for the support person since that essentially limits their ability to help you. If you're

driving or doing something else, you're often distracted and this can also delay things. Be ready to troubleshoot and preferably have some basic tools available that apply to your product.

Email Support

With email, the conversation essentially goes from full duplex to half duplex — in one direction at a time. Email has both advantages and disadvantages. One advantage is the ability to attach screenshots, code, schematics, etc. This way, the support person has these immediately. One disadvantage is that depending on the communication, it can take longer to reach a conclusion. Although, sometimes it happens quicker due to the information attached. Email is often necessary when schedules or different time zones prevent phone support. Here are some tips when submitting an email request.

Your subject line should clearly summarize your issue and/or the product you are contacting support about. The subject line can directly affect how fast you get a response. For example, when I respond to email I do it in the order it was received unless the message has come to the wrong department. These I immediately forward to the correct address. If the subject is, "Help!" or "It's not working!" it's easy to tell at a glance that these are most likely tech support messages.

Unfortunately, I will often get to these at some point during the day only to find the message is about the website or an order, and could have been taken care of faster had it been sent to the right department sooner.

In the body of your message you should clearly describe the issue, including what you have tried and the results you are (or aren't) getting. Simply saying it isn't working doesn't provide any information that can help the tech support person determine where to start dissecting your problem. Attaching photos, screenshots, source code, or schematics can help troubleshoot the issue — often without having to ask additional questions. For some things, a video is even a great help (I have used these myself).

Speaking of source code, you should always attach the program if that is what you are contacting support about. Never paste it into the message as the formatting usually gets lost, making it more difficult to spot errors. Also, when possible send only a snippet of code that duplicates the issue you're experiencing. Sending an entire application can mean hours of time pouring through code; many places won't support custom code for these reasons. However, even if they don't, some places will still look at reasonable blocks of code to solve issues. You can help the tech support person by keeping it small and in the original format.

If the tech support person replies with some questions, try to make sure you have addressed each question before replying back. I often send a series of common questions or things to try to save time, only to get a reply to just the last question, or the customer only trying the last suggestion. I then have to go back and forth several times to get all the details, only to have the customer become frustrated at the time it's taking to get their problem solved.

Check your spam filter. I have had calls from customers who assumed nobody was answering email when, in fact, the email had been answered, but ended up in their spam folder or got blocked by their company firewall. This often happens because the message contained links or a zip file or executable which is common with tech support email. Spam filters use these things in determining if a message is spam or contains malware. It could also be that somehow the sender was inadvertently blacklisted.

Forum Support

Forums are a community environment with social interaction. This, as with anything, has some advantages and disadvantages. One of the main advantages is that you are now potentially getting support from an entire community of people who probably know the product and how to troubleshoot it. I say "potentially" because it really depends on how well you can communicate your dilemma, and if anyone can answer your specific question. The odds for help go up with more participants, and you get more points of view on something than a single person can provide.

A disadvantage to forums is that as social communities, you have different types of people. A lot of times, personalities clash or people are misunderstood. Members can misread replies and assume someone is being short or abrasive with them. However, I find that on forums there are just as many people willing to help as there are that aren't. Yes, you have to have thick skin on forums, but you should not be afraid to post questions. You have to look past those who really don't want to help.

Also remember that a lot of people won't solve your issue for you necessarily, but will help you to help yourself.

As with email, a proper subject line can be the difference in getting answers to your questions on a forum. Subject lines such as, "Noob here!" often get ignored because members generally notice things they know something about. If the subject doesn't interest them, they move on.

This type of post doesn't get a lot of attention — especially on large forums with a lot of messages being posted per day. Members won't give your post attention if you don't draw them to it with a helpful subject line.

Also, you need to describe your problem in detail, including what you have tried and what the results were. The more details you provide, the more likely someone

will be able to help you because you have reduced the effort for them by providing information up front. As suggested before, attach code, provide schematics, and if you're asking for help on a product link, the datasheet (so folks don't have to go looking for info themselves to help you.

A lot of times, your particular question/problem may have come up before, so you can often get the answers you need from looking through previous posts.

When posting code in forums, you usually don't want to paste it into the message body. As with email, formatting can be lost making the code very difficult to read. Some forums have special "code tags" designed to help you paste source code and maintain the formatting.

Don't mark your post as urgent. To members of a forum, your post isn't any more urgent than anyone else's post. If you're on a deadline, then you will no doubt be told you should have posted sooner. Also, don't post your phone number or email address, or ask members to reach you outside the forums. The whole purpose of forums is that as a community, everyone benefits from the dialogue. So, members will expect you to follow up and get your replies in the same place the question was posted.

Finally, don't cross-post your support request in multiple sub-forums. Some people think this will get them help faster, but it tends to create the opposite effect as most members won't try chasing down what things were already covered in duplicate threads located elsewhere. You'll often see moderators remove or merge these threads anyway.

As a side note, if you post in a forum, you should really wait to contact tech support since multiple support techs may end up handling the same issue via different venues.

Tools of the Trade

In electronics and using microcontrollers, the chances are you have some basic tools for working on your device and/or projects. One instrument I highly recommend having is a multimeter. This one tool can help solve a lot of common issues in electronics — from verifying voltage levels to checking continuity and so on. Many multimeters have additional functions such as frequency counters, ammeters, components checkers, and more.

In the November 2014 issue of *Nuts & Volts*, I talked about the tools I commonly use on my bench. If you haven't read that article yet,

you might want to check it out. I have spent years moving back and forth from being a hobbyist to an engineering technician, and have used every tool available in my trade. Working with tools from companies such as Fluke, Saleae, PanaVise, Tektronix, and Teledyne LeCroy, I have had the opportunity to both use and review these tools and share my experiences with others.

The right tool for the job can often be the difference in whether you even have to contact tech support, and if you do you will be armed with useful information.

Final Thoughts

Basic test and diagnostic tools can make a big difference in your support experience. A positive attitude and willingness to try things out are always helpful. Patience is a must.

All the aforementioned information is the culmination of my experience on both getting and giving technical support. Perhaps we will meet at some future tech event and exchange thoughts on this subject. Or, perhaps I will have the pleasure of providing technical assistance to you. **NV**

Numbats and Morays and Wi-Fi, Oh My!

Back in the day, I decided to take on Ethernet before Ethernet was cool. I researched and studied the operation of the RTL8019 Ethernet controller IC day and night. All of that hard work resulted in the EDTP Packet Whacker, whose underlying theory is still being used out there to this day. While I was burning the midnight oil with a tray of RTL8019s, someone down the road in Palm Bay, FL was killing a candle with a relatively new concept called Wi-Fi. After signing various NDAs and begging for information, I was given access to the new Wi-Fi chipset. The new technology was only available to the public in Compact Flash packaging. Many long nights passed before the EDTP AirDrop series of embedded Wi-Fi modules finally saw the light of day and the Internet.
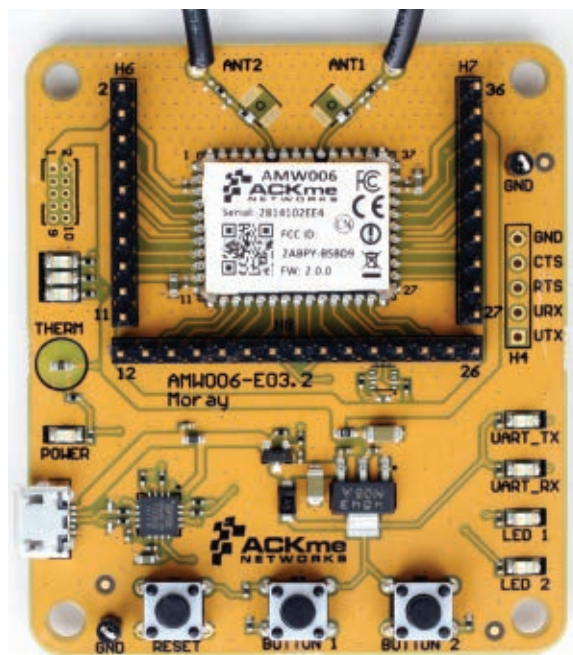
■ Photo 1. The Moray evaluation board is designed to allow the wireless designer to get the Numbat online as quickly as possible. All of its GPIO lines are attached to header pins. Easy access to the Numbat's application and configuration engine is provided via a simple USB connection to a terminal emulator running on a PC.

Back then, the initial goal was to just get the hardware to work. Then, I'd worry about how to write the driver firmware. Once all of that transpired, surely the application of the technology would be a piece of cake. As the Material Girl would say, "NOT." As it turned out, I ended up writing lots of custom API (Application Interface) and TCP/IP stack-like code to support the firmware that supported the "garage-engineered" Ethernet and Wi-Fi hardware.

Commercially available Ethernet and Wi-Fi enabled gadgets began to appear as access to the Internet became more and more the norm. No longer did the engineer have to worry about the hardware design or the TCP/IP intricacies. There had been enough pre-engineering performed to pretty much get the gadget onto a LAN right out of the anti-static package. The only hurdle was going beyond the LAN and getting the out-of-the-box "Wi-Fi Wonder" to orbit in Internet space.

Depending on your point of reference, the communicator used by the crews of the Federation is today's flip phone. The tricorder carried by Federation science officers is pretty close to what we know as a smartphone. My point is, to apply today's Ethernet and Wi-Fi technology, you don't have to know everything about how it works to make it work, even to the point of accessing the power of the Internet for monitoring and control purposes. The discussion this month will focus on a device that can — out of the box — instantly join your LAN, warp into Internet space, and perform monitor and control functions. A very capable TCP/IP stack and API are embedded into the device which is supported by a dedicated user-configurable web portal.

Animals

In my mind, the word "animals" conjures up thoughts of an album (not CD) recorded by the rock group, Pink Floyd. Do you know what a Numbat is? It's an

■ Schematic 1. The Numbat's GPIO engine employs multiplexing to squeeze in additional UARTs, SPI portals, ADC units, and PWM outputs.
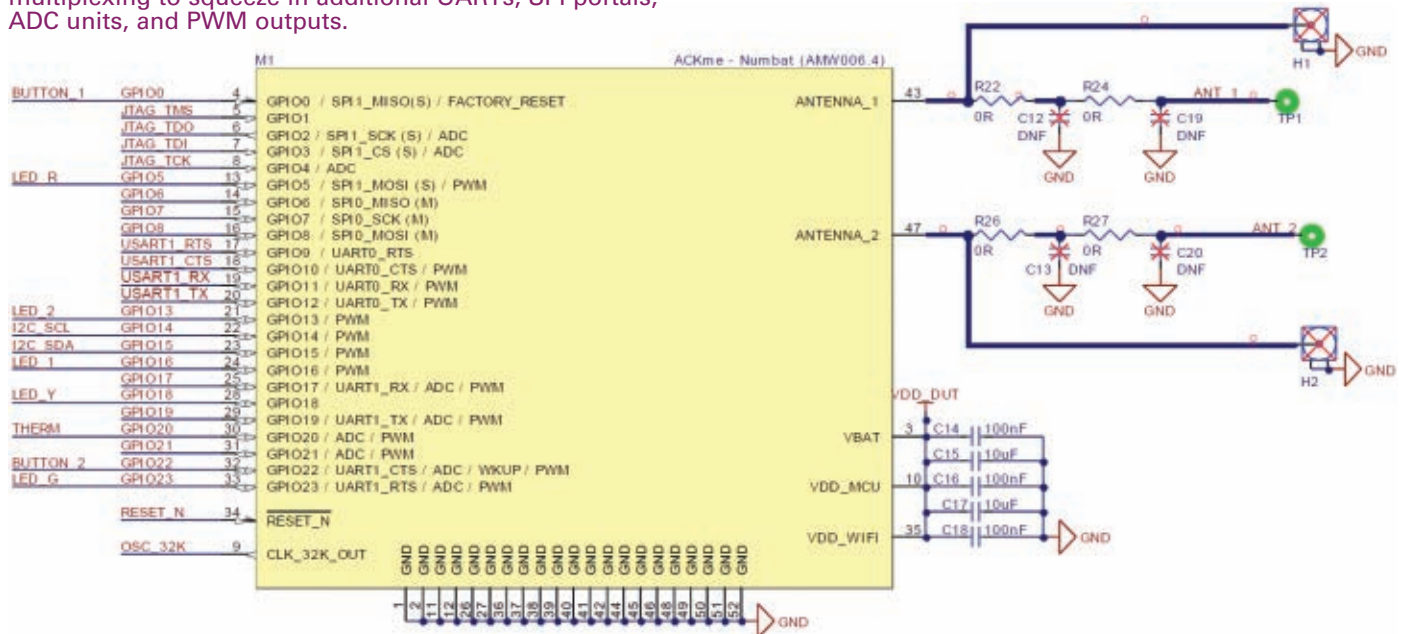
endangered species of anteater (that mainly eats termites) found in parts of Australia. The Numbat we will be concerned with is not on the endangered species list and doesn't eat termites. Our Numbat is the compute/communications engine that powers the Moray evaluation board pictured in **Photo 1**. That TCP/IP-API-web portal thing I mentioned earlier is called WiConnect.

WiConnect runs as an application within the Numbat Wi-Fi networking module. The Numbat Wi-Fi networking module is designed from the firmware up to be battery powered. This makes the Numbat perfect for remote sensor-based applications that need to transfer data over distances made possible by Internet travel.

All of the things I had to worry about when designing Wi-Fi hardware are taken care of by the Numbat engineers. The Numbat has a built-in TCP/IP stack, along with all of the modern security algorithms. In that the Numbat is driven by a low power internal MCU, it can stand alone or participate as a slave device in the network application. Note the Moray's pair of "ears" (antennae). There is a pair of aerials present because the Numbat supports antenna diversity. The Numbat is also diverse when it comes to directly interfacing electrically to locally attached peer devices.

RS-232 is common with a lot of modules like the Moray. However, the Numbat is also capable of speaking SPI and I²C. By coupling the Numbat's internal MCU and multilingual abilities with user-programmable GPIOs, timers, and ADCs (analog-to-digital converters), the Numbat can act as an independent Wi-Fi sensor node.

Animals with Teeth

That brings us to Moray, which is a nickname for an aquatic animal that dresses flashy in bright colors and has a mouth full of razor sharp teeth. The moray eel is most often depicted in movies as the underwater monster that

■ Schematic 2. This is a classic yet state-of-the-art implementation of a powered USB interface under the command of an FTDI USB-to-UART IC. Nothing gets power until everything is certified to be okay at the USB connection.

```
Ready
> get wlan.mac
4C:55:CC:10:2E:E4
> factory_reset 4C:55:CC:10:2E:E4
Reverting to factory default settings
Setting boot app to wiconnect.exe (52)
WiConnect-2.0.1.8, Built:2014-11-30 16:31:16 for AMW006.2, Board:AMW006-E03.1
[Ready]
>
```

■ Screenshot 1. The factory reset command requires the MAC address of the Numbat to assure that a factory reset is really what the user wants to do.

```
Ready
> get gpio.usage
!   # Description
#   0 Standard I/O (button1)
#   5 system.indicator.softap (softap_led)
#   6 SPI MISO
#   7 SPI CLK
#   8 SPI MOSI
#  11 UART0 RX
#  12 UART0 TX
#  13 Standard I/O (led2)
#  16 Standard I/O (led1)
#  18 system.indicator.network (network_led)
#  22 Standard I/O (button2)
#  23 system.indicator.wlan (wlan_led)
> gpio_set 13 1
Set OK
>
```

■ Screenshot 2. In this screen capture, we take advantage of the Moray's native configuration to control an LED attached to GPIO 13.

always seems to eat the bad guys. Although not as colorful, our Moray has plenty of teeth. There are 24 of them to be exact (GPIO0–GPIO23). I'm going to attach

the Moray to my laptop and check them out.

The Moray is primarily powered from its USB portal. Referencing **Schematic 1**, we can see that the Numbat and associated electronic components can be powered from the Moray's male header array. In **Schematic 2**, a workhorse LM1117 converts the USB +5 VDC VBUS signal to the Numbat's required 3.3 VDC. The Moray — being a development platform — allows the Numbat networking module to be directly powered by a battery with the removal of zero ohm resistor R1.

The POWER LED on my Moray is illuminated. So, things are good so far. Let's fire up Tera Term Pro and see if the Numbat will talk to us. I want to start with a clean slate. That means executing a factory reset as shown in **Screenshot 1**. To avoid accidentally executing the Numbat's factory reset command, the Numbat's MAC address must be entered as part of the command.

Now that we have a factory-fresh Moray, let's see just how "fresh" it is. After clearing the Tera Term Pro's display buffer, a couple of taps on the ENTER key brings the Numbat's "Ready" prompt out of hiding. To find out about the Numbat's GPIO configuration, we enter the command *get gpio.usage*.

As you can see in **Screenshot 2**, the Moray's pushbuttons, LEDs, and LAN indicators are predefined. There is also a predefined SPI portal and a simple UART interface available to us. At this point, all we have to do to illuminate LED 2 is issue the command *gpio_set 13 1*. Conversely, the command *gpio_set 13 0* will darken it.

You're probably thinking that our factory reset process was not really a true "reset." The truth is the factory reset is as fresh as you want it to be. The predefined GPIO configuration is actually the result of data contained within a GPIO configuration bootup file. The configuration file *gpio_config_init.csv* is revealed via the *ls* command in **Screenshot 3**. Taking it a cog forward, I opened the comma delimited *gpio_config_init.csv* file with the *fop* (file_open) command. The *read* command (*read* is a shortcut command for the *stream_read* command) lists the first 1,000 characters of the configuration file.

As you can see in the data listing, an alias of *led2* is assigned to GPIO pin 13. If you take another look at **Figure 1**, it's rather obvious as to how the other GPIO pin assignments relate to the file listing.

While we're talking LEDs, we can override the default

```
Ready
> ls
! #     Size    Version   Filename
# 0     1853    2.0.0     /favicon.ico.gz
# 1     18067   2.0.0     /setup/images.png
# 2     10525   2.0.0     /setup/index.css.gz
# 3     10155   2.0.0     /setup/index.html
# 4     39247   2.0.0     /setup/index.js.gz
# 5     36511   2.0.1     command_help.csv
# 6     135     2.0.0     default_setup.script
# 7     1236    2.0.0     geotrust_ca.pem
# 8     1765    1.0.0     ghm_capabilities.json
# 9     2854    1.0.0     ghm_capabilities_setup_solo.csv
# 10    277     1.0.0     gpio_config_init.csv
# 11    212736  2.0.1     upgrade_app.exe
# 12    178252  2.0.1     wiconnect.exe
# 13    203261  5.26.230  wifi_fw.bin
> fop gpio_config_init.csv
[Opened: 0]
0
> read 0 1000
16,led1,,
13,led2,,
0,button1,,
22,button2,,
23,wlan_led,system.indicator.wlan,fast_blink|medium_blink|slow_blink
18,network_led,system.indicator.network,fast_blink|medium_blink|slow_blink
5,softap_led,system.indicator.softap,static_off|fast_blink|medium_blink|slow_blink
[Closed: 0]
> >
```

■ Screenshot 3. The mere listing of the Moray's files gives us a good idea of how the WiConnect system works.

■ Screenshot 4. Short and sweet. We are using a terminal emulator running on a laptop to issue these commands. The command strings are short enough to stuff into a microcontroller's firmware, which makes the Numbat more attractive as a microcontroller-master's slave device.

```
Ready
> gdi 13 none
Set OK
> gdi 16 none
Set OK
> pwm 13 50 10
Success
> pwm 16 50 1
Success
>
```

configuration for the LED GPIO pins. Let's use the Numbat's built-in PWM (pulse-width modulation) subsystem to blink LED 1 and LED 2. The first order of business is to deregister the LED GPIO pins. This step makes sure that no other peripheral function is using the pins.

I used *gdi* shortcuts in **Screenshot 4** for the *gpio_dir* commands. Once the LED GPIO pins were cleared, I used a shortcut for the *pwm_update* command to assign 50%
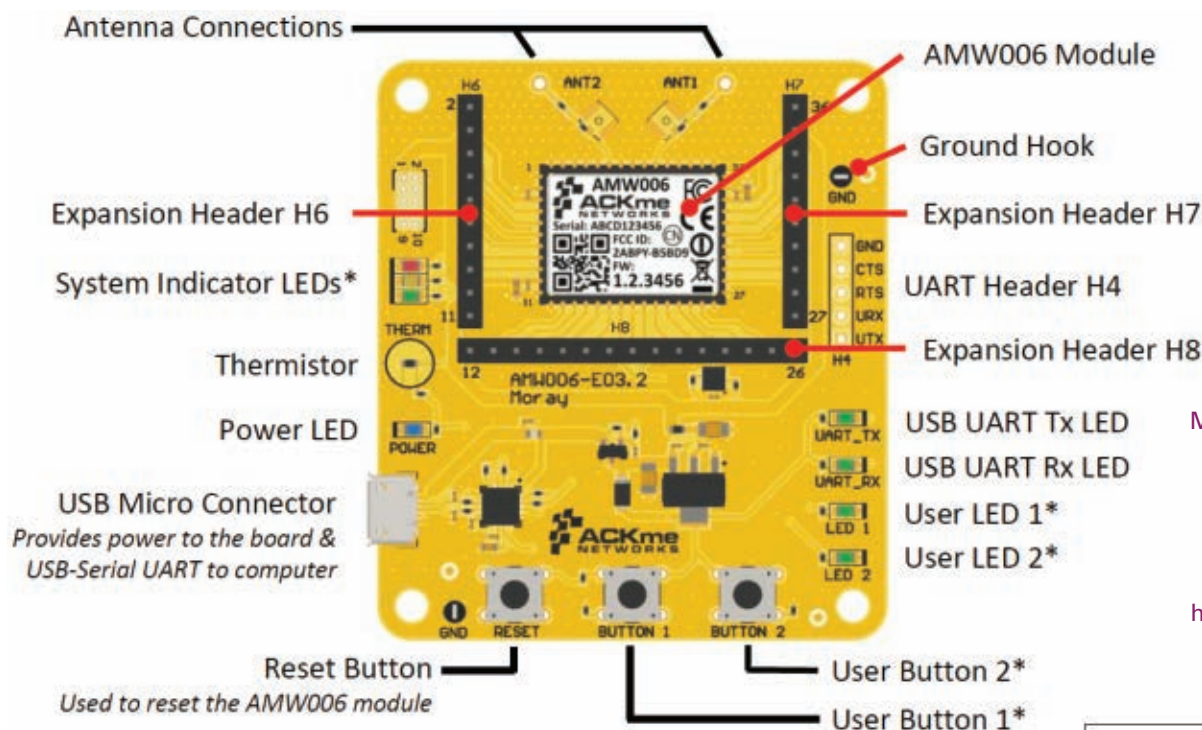
■ Figure 1. The Moray is powered from the USB connection. The only thing that may not be obvious in this illustration is the AMW006, which happens to be the endangered Australian anteater.

*User configurable

Name	GPIO
adc0	22
adc1	23
adc2	19
adc3	17
adc4	N/A
adc5	7
adc6	6
adc7	8
adc8	21
adc9	20

■ Screenshot 5. As you can see, there are nine available ADC channels we can tap using the Moray's header pins. For our purposes, I'm going to use an ADC on a GPIO pin that has not been predefined.

duty cycles to both LED pins. The LED attached to GPIO pin 13 will blink at 10 Hz, and the LED controlled by GPIO pin 16 flashes at 1 Hz. We're just blinking LEDs here because that happens to be what the GPIO pins are driving. The PWM waveforms I dialed up with the commands can be applied to anything that may be hanging on those pins.

Not all things are digital. A PWM signal may be just the ticket for the speed component of a motor controller. However, there are some data elements that only show themselves as voltage. It's so easy to use the Moray's ADC engine, I won't have to show you a graphic describing it.

Let's say we need to monitor the voltage of a battery that is critical to our application. Checking **Screenshot 5**, we find that the Numbat has a number of potentially available ADC channels. According to the Moray GPIO layout diagram in **Screenshot 6**, ADC8 is sitting on an unassigned GPIO pin and is free to use.

To get a voltage reading from ADC8, all we have to do is execute the command *adc 21*. A 12-bit hexadecimal ADC value will be returned. The returned hex value represents the measured voltage in millivolts.

I dialed in +1.2 VDC on my bench supply and attached the power supply leads to the Moray's GPIO21 header pin and ground ring. Following the entry of the command *adc 21,* I received a return hexadecimal value of 0x5FA (1530 decimal).

To convert the raw hex value to millivolts, I used this formula provided by the Moray documentation:

```
adc_mV = (Vref * adc_value) / adc_max_value
```

Where:
Vref = 3,300 mV
adc_value = 1530 (steps)
adc_max_value = 4095 (steps)
1,232.9 mV = (3300*1530) / 4095

There is only one decimal place resolution on the power supply indicator. So, my finger wobble most likely accounts for the extra 32.9 mV.

GPIO14			pwm3		22	none			H8-22
GPIO15			pwm3		23	none			H8-23
GPIO16			pwm6		24	Standard I/O	led1	User LED1	H8-24
GPIO17	uart1.rx	adc3	pwm4		25	SPI CLK			H8-25
GPIO18					28	system.indicator.gpio network	network_led	LED Y (Yellow)	H7-28
GPIO19	uart1.tx	adc2	pwm4		29	SPI MISO			H7-29
GPIO20		adc9	pwm2		30	none		Thermistor	H7-30
GPIO21		adc8	pwm2		31	none			H7-31
GPIO22	uart1.cts	adc0	pwm1	wake	32	Standard I/O, wake, web setup	button2	Button 2	H7-32
GPIO23	uart1.rts	adc1	pwm4		33	system.indicator.gpio wlan	wlan_led	LED G (Green)	H7-33

■ Screenshot 6. The Moray's onboard thermistor is permanently attached to GPIO20. However, GPIO21 is free for the taking as ADC8.

Catch and Release

We have found our Moray to be more than capable of measuring voltages, generating PWM signals, and processing digital I/O.

It's time to take this show on the road and expose our Moray to the information ocean we know as the Internet. We will begin by getting our Moray up on the local EDTP LAN.

As you have seen thus far, the Moray is super easy to use. Getting our Moray up on a LAN requires that we know the SSID and passkey of the LAN we wish to join. Once we have SSID and passkey information, there are three simple steps we need to execute to allow our Moray to join that particular LAN. Here's what the sequence looks like:

```
Ready
> set wlan.ssid edtp
Set OK
> set wlan.passkey secretkey
Set OK
> ping -g
[Associating to edtp]
Security type from probe: WPA2-AES
Obtaining IPv4 address via DHCP
IPv4 address: 192.168.0.103
[Associated]
Ping reply in 2ms
>
```

Any command requiring network access following the pair of *set wlan* commands will force the association process you see in the captured terminal text. Ping works just fine here as it requires a reply from an ICMP server up on the network.

Okay ... We're Up on the LAN

Now what?? Normally, we would have to set up some sort of a server on our LAN that we could gain access to via the Internet. That involves using a DNS service to point at a particular piece of server hardware operating behind a selected IP address on our local router.

If you're a regular Design Cycle reader, you've been involved with the DNS process before. I'm sorry. We don't have to do any of that server-router-DNS-service-setup stuff to get our Moray on the Internet. If you have access to a Moray, you have access to the Internet via **https://gohack.me**.

Let's walk through accessing the **gohack.me** site. I already have an account. So, we can start with **Screenshot 7**, which will allow us to activate our Moray device. Since we're going live for the first time, we will follow the script and get the canned capabilities files from the **gohack.me** server.

We are already up on a LAN. So, to get the capabilities files, we simply enter *ghm_capabilities download –s* in our Tera Term Pro window. The *–s* indicates that we will be running in SOLO mode.

In SOLO mode, the Numbat is in stand-alone mode and does not answer to an external master microcontroller. The capabilities download process looks like this from Tera Term Pro's perspective:

```
[Ready]
> ghm_capabilities download -s
[Associating to edtp]
Obtaining IPv4 address via DHCP
```

■ Screenshot 7. Once you purchase an ACKme device such as the Moray, you are entitled to a spot on the **gohack.me** site. The idea is to give you easy access to the Internet for testing your ACKme device application.

```
IPv4 address: 192.168.0.103
[Associated]
Request GET
/2.0/ghmcaps/amw006_e03.1/ghm_capabilities?do=
    export_raw
Connecting (http): wiconnect.ack.me:80
HTTP response: 200
Capabilities file: ghm_capabilities.json created
Erasing duplicate file: ghm_capabilities_
    setup_solo.csv
Request GET /2.0/ghmcaps/amw006_e03.1/
    ghm_capabilities_setup_solo?do=export_raw
Connecting (http): wiconnect.ack.me:80
HTTP response: 200
Setup file: ghm_capabilities_setup_solo.
    csv created
Running setup script:
ghm_capabilities_setup_solo.csv
```

The capabilities download includes a script file that sets up the Moray GPIO. The script kicks off the command to show us what it has done for us:

```
# GPIOs configured for use with goHACK.me ...
get   ghm.solo.gpio all
!   # Description
#   0 stream.button1 - in,edge-rising
# 13 control.led2-duty - pwm
# 16 control.led1 - out
# 19 control.gpio19 - out
# 20 stream.thermistor - adc
# 21 stream.gpio21 - inpd
# 22 stream.button2 - in,edge-rising
```

Everything is covered in the Moray GPIO configuration. All that is left to do now is activate our Moray device:

■ Screenshot 8. All of the Moray's configured GPIO is represented graphically and visually without us having to write any HTML code or configure any DNS communication links.

```
> ghm_activate fred@edtp.com secretkey
Request POST /api/activate/login
Connecting (https): api.gohack.me:443
Starting TLS
HTTP response: 200
Starting goHACK.me solo mode
[2014-13-67 | 05:03:30: Disassociated]
Rebooting
[Associating to edtp]
WiConnect-2.0.1.8, Built:2014-11-30 16:31:16 for
    AMW006.2, Board:AMW006-E03.1
[Ready]
> [Associated]
```

Turning our attention from the Tera Term Pro session and back to the **gohack.me** site, we find that the Moray's GPIO configuration maps directly to the contents of the **gohack.me** page captured in **Screenshot 8**. As you can

see in the web page screen capture, I've punched both pushbuttons, turned on LED 1, and set LED 2 blinking at 1 Hz.

Getting PICy with the Moray

The Numbat is a very capable piece of network hardware. We've seen what it can do without any help. However, in our world, it isn't always practical to push commands to an embedded network device over a laptop's USB port. So, in the next installment of Design Cycle, we will use the CCS C compiler and a tricky PIC microcontroller to control the Moray's data stream in slave mode. **NV**

ELECTRONET

CubeSats — Part 2: The Basic Subassemblies

Last time, we discussed the invention and growing use of CubeSats. I hope this whet your appetite because this month introduces some of the kits and basic subsystems that are available for those who want to start their own space program.

Available CubeSat Kits

Wanting to know who actually produced CubeSat kits, I did a quick Internet search and turned up two major manufacturers: Pumpkin (**www.pumpkininc.com**) and Clyde Space Ltd (**www.clyde-space.com**).

A later search for subsystems turned up a third manufacturer of CubeSat kits: Innovative Solutions in Space (ISIS; **www.isispace.nl**). Here's what I discovered about CubeSat kits from these three companies.

First is Pumpkin; they're part of Salvo, the manufacturer of a Real Time Operating System (RTOS) for single-board computers (SBC) and microcontrollers like the 8051, ARM, Atmel, MSP430, and M68HC11. Pumpkin's

kits are distributed by the company, CubeSat Kit (**http://cubesatkit.com**).

CubeSat Kit sells an entry-level pack (part number 711-00227) or barebone's CubeSat. This kit doesn't contain everything for a flight-ready CubeSat. You'll still need to add the following subsystems: power, telemetry, and sensors. However, the kit does contain the following items:

- Solid-wall cubic airframe (10 centimeters on a side)
- MSP430 SBC flight computer loaded with the Salvo RTOS
- **Remove Before Flight** pin to safeguard the CubeSat before it's loaded into the P-POD (Poly-Picosatellite Orbital Deployer)
- Programming software and programming cable
- Five volt power supply to power the CubeSat on a workbench

The basic kit can be customized with different airframes and flight computers. The available airframes

(C) Copyright 2003 Pumpkin, Inc.

It's like getting a Christmas present. These are the parts you get when you purchase the Pumpkin CubeSat kit.
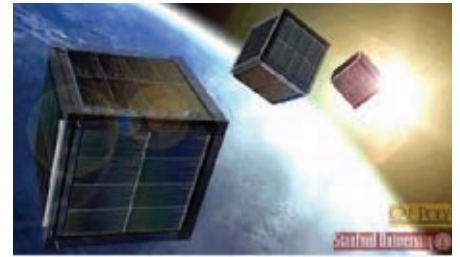
include those for a 0.5U, 1.5U, 2U, and 3U CubeSat, and the flight computers include Atmel and M68HC11 based PC/104 cards.

As mentioned, the second company I found is Scottish Clyde Space Ltd. Clyde Space doesn't just sell CubeSat parts, they also sell subsystems for larger satellites. Unlike the company CubeSat Kits, Clyde Space doesn't sell a complete kit. Instead, they sell the subsystems needed to construct a CubeSat. You order their subsystems online almost like a Chinese restaurant menu by selecting a particular version of each subsystem.

Clyde Space's categories of subsystems include: Electrical Power System (EPS), Power Distribution, Batteries, Solar Arrays, Deployable Arrays, Attitude Determination and Control Systems, De-orbit Devices, Propulsion, Structures, Communications, Onboard Computers, Software, Ground Support, Ground Stations, and Accessories.

The last company I found (ISIS) is based in the Netherlands. ISIS produces a complete 1U CubeSat kit, but what I found even more interesting was their CubeSat engineering model. This model is a training tool that allows anyone to build and test hardware and software for a CubeSat.

The kit includes a 2U airframe, flight computer, EPS, and telemetry. To create a more realistic environment, the

Post comments on this article and find any associated files and/or downloads at
www.nutsvolts.com/index.php?/magazine/article/february2015_NearSpace.

The CubeSat engineering kit from ISIS comes with enough parts to create a functioning model of a CubeSat on your electronics bench. Even better, you can use the hardware and software that you develop to make a flight-ready CubeSat.

This CubeSat was tested on a high altitude balloon prior to its launch. Since this CubeSat has no solar arrays, you can see the skeletonized panels making up its airframe. *Image from NASA.*

Eight metal clips attached to the top and bottom corners of an airframe provide support for a CubeSat's solar panels. *Image from CubeSat Kits.*

development kit includes a ground station, so you communicate with the CubeSat engineering model over its UHF radio.

Lots of CubeSat Parts

Since CubeSats are mission-specific, builders shop around for their subsystems, or learn to design and make the systems themselves. Before designing parts for yourself, however, you should become acquainted with some of the available subsystems.

There are lots of retailers of CubeSat subsystems like Clyde Space, but one retailer you might want to check out is CubeSat Shop (**http://cubesatshop.com**). Here are some examples of the basic subsystems I discovered online.

Airframes and Solar Panel Clips

Without an airframe, a CubeSat is just a stack of PC/104 cards.

Therefore, an airframe kit is one of the first things you need to purchase (if you didn't purchase a complete CubeSat kit).

I discovered that CubeSat airframes are assembled from aluminum panels that are then bolted together. The size of the panels depends on the final volume of the CubeSat. Whether it's 1U or 3U, the width of airframe panels is always 10 centimeters. The length of the panels comes in units of 10 centimeters, depending on the volume of the CubeSat. For example, a 1U requires a 10 cm long panel, whereas a 3U requires a 30 cm long panel.

Airframe panels are available in two forms: solid aluminum sheets or in a skeletonized form that removes 68 grams of metal per panel for a 1U CubeSat.

Most CubeSats get operating power through solar panels. However, solar panels face two risks if they're bolted to a CubeSat

airframe. First, is the acoustical vibration the CubeSat experiences during its ride to earth orbit; rocket engines are loud and will rattle things loose.

The second risk comes from the contraction and expansion that a CubeSat experiences every time it passes in and out of the earth's shadow. Since the aluminum airframe and silicon solar arrays contract and expand at different rates, passing into and out of the earth's shadow can produce stresses that will crack the delicate solar panels.

Attaching the corners of the solar arrays to the sides of the CubeSat airframe with clips eliminates these risks. The eight aluminum clips that do this bolt to all eight top and bottom corners of the airframe.

The clips are folded in such a way to create pockets that gently

An example of a single-board computer in the PC/104 form factor. This version is available from CubeSat Shop. *Image from AMSAT UK.*

Two examples of CubeSats with deployed arrays. *Image from Clyde Space.*

secure the corners of the solar arrays in place.

Controllers and Single-Board Computers

The Salvo RTOS is available for many types of microcontrollers and SBCs. Since all use the standard PC/104 bus, it's really simple to install any one of their flight computers into a CubeSat. In addition, Salvo has an option to purchase just a motherboard which lets you select an SBC at a later time, or upgrade an older SBC.

Power

Some CubeSats only use a primary battery for power and therefore have a limited lifetime in orbit. Most CubeSats, however, use a combination of solar cells, charger, and batteries. Normally, 1U CubeSats just mount solar cells to their exterior surfaces with clips. Some 1U and larger CubeSats, on the other hand, deploy unfolding solar arrays after ejection from the P-POD.

The benefit of deploying arrays is that a CubeSat has larger arrays and more power. Deployed arrays attach to the top of the CubeSat airframe with springs and hinges. The folded array attaches to the bottom of the airframe with a plastic cord. Once in orbit, a hot wire cutter melts the cord and releases the solar array.

At the end of their deployment, the spring-mounted solar arrays cushion themselves so that they gently finish their extension rather than slamming to a stop.

The charging of CubeSat batteries is managed by the CubeSat's EPS, or electrical power system. The EPS is a single PC/104 board that interfaces the batteries

A LiPo battery within a PC/104 form factor. Plug this in and your CubeSat has power. *Image from Clyde Space.*

and solar array to the electrical system of the CubeSat. An EPS controls battery charging and regulates battery voltage. Many provide both 5.0 and 3.3 volts to the PC/104 bus running through the CubeSat.

CubeSat batteries are either rechargeable lithium-ion or lithium polymer (LiPo) batteries. Using lithium-based chemistry is a necessity for CubeSats since they are severely weight and volume constrained. All commercially-available CubeSat batteries are tested for capacity, reliability, and the ability to function properly in a vacuum.

Telemetry

A CubeSat mission is useless unless it can communicate with the ground. Most — if not all — CubeSats communicate using amateur radio frequencies. Therefore, a CubeSat owner must be a licensed ham radio operator.

(Not a ham radio operator? No problem! *Nuts & Volts* has a new column on ham radio that appears in the odd months ... check out the intro column in January's issue, then stay tuned for the March edition!)

Because of a CubeSat's small size, their antennas are necessarily short. Short antennas work best at high frequencies, so this makes VHF, UHF, and S-band

microwave the most useful frequencies for CubeSat telemetry.

A popular mode of amateur radio communication is the automatic packet reporting system, or APRS. Many CubeSats use it for their telemetry since satellite status and science data is easily transmitted to ground stations using this mode.

In addition to sending data to the ground and receiving commands from the ground, many CubeSat radios beacon their identity using Morse code. A CubeSat radio only needs to transmit at a couple of watts because that's enough to be heard with a handheld amateur radio connected to a directional antenna like a yagi.

However, a correction for frequency changes due to Doppler shift is necessary because of the orbital speed and changing direction of travel of the CubeSat. So, incorporating a frequency agile radio and controlling software is useful.

So, what do CubeSats use for their VHF and UHF antennas? Would you be surprised to hear they use a metal tape measure?

CubeSats use them because a tape measure is flexible and can roll up into a small loop. It then springs back into shape when released (as long as it's not bent). Since everything is weightless in space, antennas remain perpendicular to the CubeSat. This property makes metal tapes an excellent material for CubeSat antennas.

One interesting antenna solution I found is the Deployable Antenna System from ISIS. This product lets you mix the antenna elements. This means a CubeSat could have four separate monopole antennas for four different frequencies, or two dipoles for different frequencies like VHF and UHF radios.

The last CubeSat antenna

This is the ISIS VHF/UHF transceiver radio. It uses a VHF frequency to transmit to ground stations, then ground stations use a UHF frequency to transmit to the CubeSat. The radio's specific VHF and UHF frequency is determined by crystals installed on the printed circuit board.

solution is for the microwave S-band transmitters. Clyde Space sells a three inch diameter patch antenna that attaches to the earth-facing (nadir facing) side of a CubeSat. The antenna can handle two watts of power, and has a beam width wide enough that the CubeSat doesn't need to be pointed very accurately towards earth.

That's the basics of a CubeSat, airframe, computer, power, and telemetry. Next time, we'll dive into

some of the ways CubeSats control their orientation in space (this is called attitude), and some of the sensors currently available. As I research CubeSats for this column, I keep getting surprised with what's available. I think you will too.

Onwards and Upwards,
Your near space guide **NV**
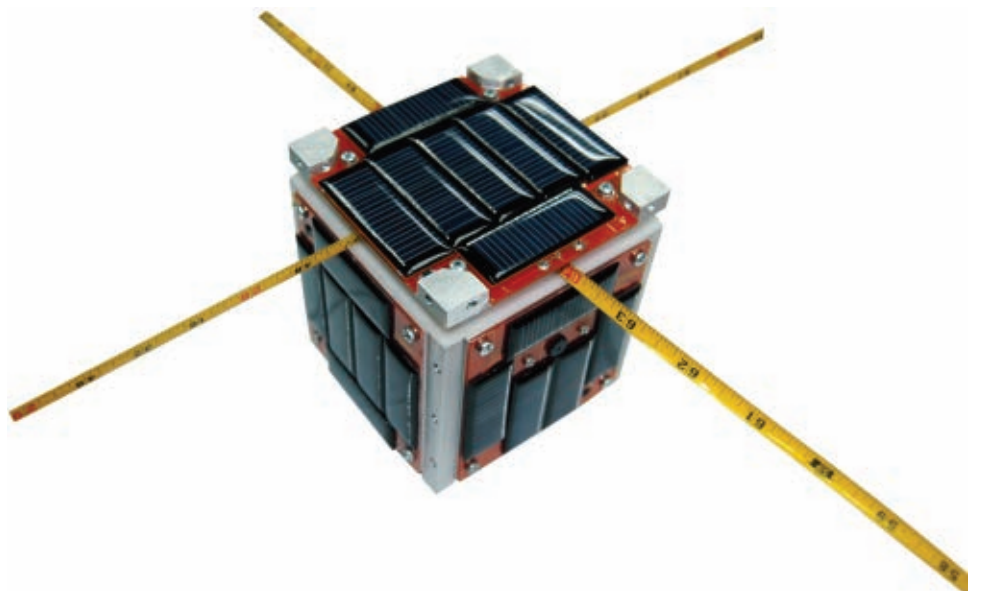
An AMSAT UK CubeSat displaying its deployed antennas. The antenna elements wrap around a slot in the top of the CubeSat prior to launch.

A Custom 3D Printed Case for CHIPINO

Designing a custom case for an electronic module often includes finding an off-the-shelf box, designing the circuit board for the box, and then possibly getting cut-outs professionally done by the box supplier or the old drill and saw method of making openings. With 3D printing, all that has changed. You can now design the case you want from scratch (or from somebody else's design).

*3D printers are the wave of the future — or so countless articles, reviews, breathless news commentators, and, of course, the machine's manufacturers keep telling us. By now, we've all seen a plethora of itty-bitty cubes, Yoda heads, chess pieces, interlocking gears, and other interesting but ultimately useless "things" created with 3D printers. Though many of these little demo pieces are impressive by themselves, they never quite cross over into the realm of *useful.* As we believe the "what is 3D printing" topic has been done to death, we thought it was high time to bring you a useful series on how to actually implement 3D printing. Specifically, working with 3D printers and showing you how to use them for practical projects on your workbench.*

I've built cases in the past that were designed to be cut from flat stock and then assembled. That required me to find a CNC cutter, or cut them out by hand on my scroll saw. Fortunately, I have a local source with a laser cutter to make the job easier. However, I'm always looking for ways to do something other than a flat wall square corner box. I wanted to make a unique case for a CHIPINO module.

The CHIPINO module is built similar to an Arduino but runs on a Microchip PIC device. This allows me to program in PICBASIC or Great Cow Basic, or even C if I want. A search on **Thingiverse.com** for unique Arduino style cases revealed some interesting options, but nothing that fit what I wanted. So, I then searched the designs at **Tinkercad.com**. This is the online software I use to create my 3D designs; they have a gallery of user-released designs. One caught my eye by user Agustin Seville (**Figure 1**).

It was a square case with a bubble-like edge. The best part is it was released open source in Tinkercad, so I could modify the original source file as long as I gave Agustin original attribution and released my design open source, as well. I went to work to create my custom version.

Tinkercad makes designing as easy as building with LEGOs. You just modify the design by placing blocks of different shapes on top of each other. The best feature is the ability to make any block a hole. This means whatever it touches takes material away. I wanted a rectangular hole in the top so a CHIPINO proto shield could pop through, so that just required a rectangular box placed through the top as a hole.

The original design has built-in snaps. I like that because no screws were required to assemble it. I left that part of the design untouched. I did have to modify the front hole that surrounded the USB connector on the Arduino. The CHIPINO has a programming header for connecting to a PICkit programmer, so I once again used a block shape set as a hole to make the opening wider. I also removed the block shape hole that was part of the original design. That block created the USB hole in the first place.

The mounting tabs matched the back holes of the CHIPINO perfectly, but the side tabs were off a little. I

■ **FIGURE 1.**
Arduino case by Agustin Seville.

Post comments on this article and find any associated files and/or downloads at **www.nutsvolts.com/index.php?/magazine/article/february2015_Practical3DPrinting**.

adjusted those and the bottom half was set. I printed what I had to see how everything fit. I used a 0.4 mm layer height for the print so it would be a bit rough but much quicker. Then, I placed a CHIPINO (with a proto shield on top) into the box. I found I had a few tweaks to make.

The reset switch was hitting, and also the edge of the board was pressing against the inside rounded edge. Once again, I used a hole shape to put a round hole at the switch location and a square shelf in the rounded edge for the board to fit. Another print at the 0.4 mm layer and the test unit was ready. It fit quite nicely.

I then decided to change a few things inside. The original design had a Tinkercad symbol on the inside bottom, so I changed that to text that was about 1 mm tall that spelled out **chipino.com**. There was a locating post at the back of the base that would interfere with the CHIPINO board. Honestly, I don't think an Arduino would fit around that locator properly — at least not the older style one I had in my pile of boards. So, I removed that locator and the snaps still held the board and case together just fine.

I printed a new design at a 0.2 mm layer height with a 50% fill so everything would be strong, including the snaps. I had actually printed the test units at 10% fill and the snaps broke off easily. I also printed it with supports so the rounded walls were properly supported during the print. Supports are just extra plastic that gets printed under overhangs so the plastic doesn't sag while still hot. Supports are designed to breakaway easily.

I used my Davinci 1.0 3D printer to make all the designs, and it delivered as usual with great prints. (As I've said before, it's the best $500 I've spent on my electronics lab in a long time.) The final design still had some rough surface area on the rounded walls, but that was mainly from the breakaway supports leaving a little plastic behind. A little trick is to use acetone on a cotton swab or cloth to rub out the rough sections. The acetone — commonly used as nail polish remover —
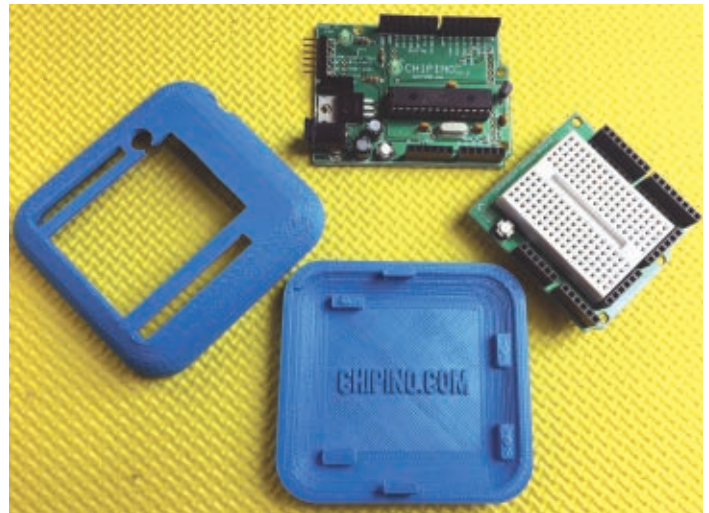
■ FIGURE 2. CHIPINO case.

melts the plastic to a softness that allows you to rub it smooth, then the plastic hardens again smoother when the acetone evaporates.

This is just the start of what I can do with my Davinci 3D printer and Tinkercad software. I hope to design some other custom cases from scratch in the future. I also created a YouTube video showing this project. You can see it at **www.youtube.com/user/beginnerelectronics**.

I have many more 3D projects there if you are interested in seeing what I'm working on next. If you have a 3D printer question or project idea, send me an email at chuck@elproducts.com and I'll try to help. **NV**

Resources

Check out my website and blog: **www.elproducts.com**

Check out my YouTube Channel: **www.youtube.com/user/beginnerelectronics**

Check out my 3D designs:

www.thingiverse.com/elproducts/designs

Tinkercad: **www.tinkercad.com**

Davinci 3D: **us.xyzPrinting.com**

Davinci Forums: **www.solidforum.com www.voltivo.com**

■ FIGURE 3. Finished design before acetone smoothing.

PROJECTS

Accelerometer Kit

The Hockey Puck accelerometer is a compact digital G-force recorder. Find out if your packages are being handled with care (or not!) by your shipping company. See if your crated animals are being well treated on airplane flights. See how many Gs are being applied to the groceries in the trunk of your car. Tie it to the end of a string and see how many Gs you can generate swinging it by hand.
**$59.95**

Neon Transistor Clock Kit
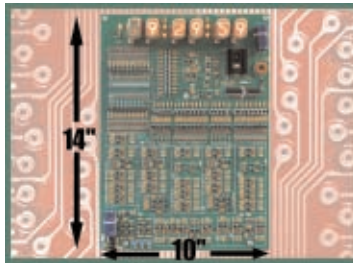
Add HIGH VOLTAGE to your clock!
This is a Nixie Tube display version of the Transistor Clock. It uses only discrete components — no integrated circuits. Approximatly 1,250 of them!
This kit is a "must have" for the analog electronic hobbyist.
For more info, see the April 2012 issue or download the 87 page manual from our webstore.
Sale Price **$245.95**

3D LED Cube Kit
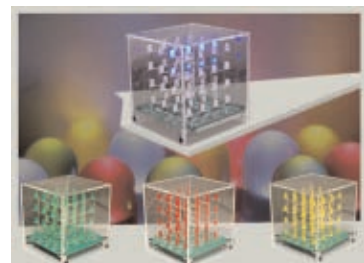
This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes.
Colors available: Green, Red, Yellow & Blue

$57.95

Solar Charge Controller Kit 2.0

If you charge batteries using solar panels, then you can't afford not to have them protected from over-charging. This 12 volt/12 amp charge controller is great protection for the money. It is simple to build, ideal for the novice, and no special tools are needed other than a soldering iron and a 9/64" drill!
$27.95

Geiger Counter Kit

This kit is a great project for high school and university students The unit detects and displays levels of radiation, and can detect and display dosage levels as low as one micro-roentgen/hr. The LND 712 tube in our kit is capable of measuring alpha, beta, and gamma particles.
Partial kits also available.
$159.95

Super Detector Circuit Set

Pick a circuit!
With one PCB you have the option of detecting wirelessly: temperature, vibration, light, sound, motion, normally open switch, normally closed switch, any varying resistor input, voltage input, mA input, and tilt, just to name a few.
$32.95

FOR BEGINNER GEEKS!

The Learning Lab 1
Fundamental Concepts
$59.95

The Learning Lab 2
Basic Digital Concepts and Op-Amps
$49.95
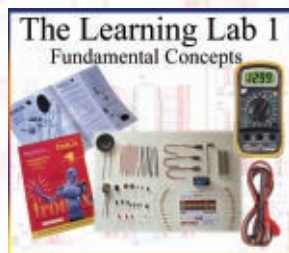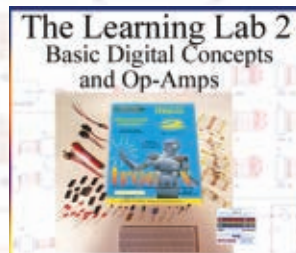
The Learning Lab 3
Basic Electronics: Oscillators and Amplifiers
$39.95

These labs from LF Components show simple and interesting experiments and lessons, all done on a solderless circuit board.

As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal

of knowledge with each successive experiment.

For more info and lab details, please visit our webstore.

>>> QUESTIONS

Obsolete IC Hunt

I ran across an old issue of a hobby magazine which had an article "Time On Your Hands" November-December issue circa 1973. I am trying to find a source for the Mostek MK5017A BB which is used in this article. Does anyone know where I might purchase a couple of these chips at a reasonable price?

I have spent several hours on the Internet and have not been able to locate a supply from any other place. Maybe one of your readers would have some of these they might be willing to part with for such a good cause.

**#2151   Ed Edmondson, PhD
via email**

Battery Dilemmanator

About three years ago, I put together a 1.5 volt battery eliminator using a wall wart feeding into an LM4120 regulator. My goal was to power the clocks I have around the house and save myself the aggravation of replacing the batteries all the time. The clock that I started with is a Howard Miller mantle clock that a company awarded to me for busting my ass for 25 years.

After installing the eliminator, I set the clock to the time of my crack atomic wristwatch and let'er go. The clock ran for almost three years with phenomenal accuracy, matching my watch within a few seconds (a hex on those who disbelieve this). The clock finally died — probably from exhaustion — having gotten no rest between battery exchanges.

Well, I thought, what are you waiting for. Get with it with the other cheap clocks cluttering up our house; so, I did. To my amazement, none of the clocks running on the eliminator could keep time anywhere near what could be termed accurate — no matter how much I adjusted the voltage (usually, the clocks ran fast).

So, what gives? Why does a battery work and my eliminator won't? Some wizard out there must know what the problem is, and be willing to share the knowledge with me.

**#2152   Joe Graf
via email**

Diode Decision

L1 is 25-50 turns of 30 gauge magnet wire.
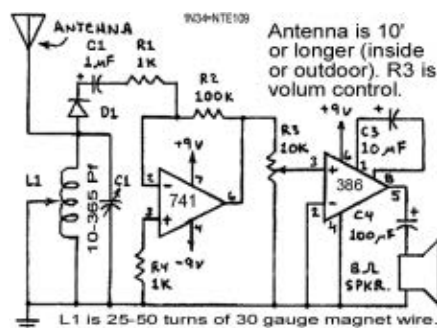
Can you PLEASE indicate which germanium diode would best fit this SW radio? Either a 1N34A or 1N60?

**#2153   Michael Williams
Crete, IL**

>>> ANSWERS

[#1151 - January 2015]
IC Identification

Figure 1 (see Jan 2015 issue) is the front-end of a simple frequency counter 1 Hz to 1 MHz. I can't seem to find IC1 4583 (I'm guessing it's a CD4583). Pin 4 of IC1 feeds a CD4026. Q1 is a 2N930 and IC5A is a 556.

I found two conflicting datasheets on this. One says it's a microcontroller (which I doubt); the other says it's a flip-flop. I've looked at Mouser and Digi-Key and can't find this IC.

#1 Because most of the other ICs in this circuit are standard 4000 series CMOS parts, that indicates the 4583 at IC1 likely is also. Using that to refine the search, the part is a dual Schmitt trigger buffer.

The unique property of a Schmitt trigger gate is that it provides hysteresis. When a slow rising, slow falling, or noisy analog signal is applied to an ordinary gate, that gate may oscillate and switch several times as the input crosses its switching threshold. Because this Schmitt trigger requires a much higher input voltage to switch high than it does to switch back low, it will clean up its analog input, and its output will switch only a single time.

In this circuit, that behavior is important because the input signal is being used as a clock for the counters. Without the Schmitt trigger, some input signals might trigger multiple clock pulses per cycle, resulting in wildly incorrect frequency numbers.

Searching for Motorola's specific version number of this part (MC14583BCP) seems to produce better search results, revealing data sheets and even a seller of the part on Amazon. However, because the circuit does not actually take advantage of the unique programmable hysteresis feature of the 4583 (by just tying pins 5, 6, and 7 together), any other CMOS Schmitt trigger should work just as well, if you can find something cheaper and don't mind slightly modifying the circuit.

**Richard Carlson
Fort Collins, CO**

Send all questions and answers by email to **forum@nutsvolts.com**
or via the online form at **www.nutsvolts.com/tech-forum**

#2 I found an MC14583B IC in a 1991 Motorola data book. This part is a dual Schmidt trigger IC and appears to match the pinout shown in the schematic. The pinout is not fully legible in the electronic issue. Using a Schmidt trigger in this application makes sense.

If IC1 is indeed an MC14583B or equivalent it should not be powered from 9V! The logic high output from Q1 is only 5V. The datasheet for the MC14583B indicates the minimum input voltage required to guarantee a logic high to be 80% of its supply voltage. Also, the logic high output voltage from IC1 is too high for the 5V downstream logic.

It is possible that IC1 is not actually defective, but is not seeing a logic high due to poor design. This theory can be tested by substituting a seven volt power supply for the battery. If it works at the lower voltage, I would modify the circuit to power IC1 from 5V.

I also question not having a resistor between the junction R1/C1 and the base of Q1 to limit input current to a non-destructive level! This might best be accomplished by cutting a 1/16" gap in the track leading to the base of Q1 and soldering a 1K surface-mount resistor across the gap.

Dick Pope
Eatonville, WA

#3 The 4583 IC in your frequency counter is a non-inverting dual Schmidt trigger. I found this in an old Motorola CMOS data book from 1991 #Q4/91 DL131 Rev 3 on page 6-498. The Motorola part number is actually MC14583B. Only the A Schmidt is used in your circuit; the B Schmidt is unused. The Schmidt trigger is used in front of the counter IC1 to ensure a clean input to the counter. Typically, the Schmidt trigger output will transition low when the input drops below 40% of the supply volt-

age, and the output will go high when the input goes above 60% of the supply voltage. For the MC14583B device, these thresholds are adjustable via external resistors through the positive, negative, and common terminals. In your circuit, those three pins (5, 6, and 7) are bussed together without a resistor and this sets the thresholds to around 40% and 60% of the supply voltage.

Pin 9 is the A input.

Pin 4 is the A output.

Pin 15 is the B input (unused, so tied high to 9V).

Pin 10 is the B output (unused).

Pin 5 is the negative A (pins 5, 6, and 7 are tied together to set the A device threshold).

Pin 6 is the positive A.

Pin 7 is the common A.

Pins 1, 2, and 3 are common, positive, and negative for the B Schmidt (unused).

Send all questions and answers by email to **forum@nutsvolts.com**
or via the online form at **www.nutsvolts.com/tech-forum**

Pin 16 is VDD. Supply voltage is 18 volt max for this device.

Pin 8 is VSS. Supply voltage return.

#4 A replacement for the '4583' could be the MC14093B or CD4093 available from Newark, etc. Although these do not have adjustable thresholds, they will be close to those of the 4583. These are quad two input NAND Schmidt triggers, so all unused input pins must be tied high (to 9V in your case).

Roger Baker
Redmond, WA

[#10131 - October 2013]
Moving to SMT

I want to move to SMT components. Do I need a hot air reworking station, or just a hot air station? Can you recommend something for someone with a small budget?

I have had very good luck using "regular" soldering techniques with the correct solder and flux. Set the component in place, then solder pins on diagonally opposite corners. If it's not straight enough, straighten it out while there are only two pins soldered.

The link I've included shows the result from using a toaster oven — with modifications to solder a whole board at one time. There are a number of videos on YouTube showing how to convert a toaster oven.

The options are to manipulate the temperature profile manually, or you can use a thermocouple and a small computer such as an Arduino to automate the temperature profile.

https://www.youtube.com/watch?v=_5lksMvmqQc

Of course, I'm still doing the soldering manually.

Ray Perry
Jacksonville, FL

Tech-Tip
Good Method To Cut Traces On A Stripboard

I have seen that people use a knife to cut and remove a piece of the copper strip on a stripboard PCB *(example N&V Jan 2015 Issue, page 16 picture 8)*. I use a 1/8 or 5/32 drill bit to drill out the copper around a hole. It is very clean and only takes two seconds. I put shrink tubing over the drill bit so as not to cut myself. Only the tip of the bit is exposed. Hold the bit as you would hold a jeweler's screw driver, put it over a hole, and with little pressure give it a twirl or two clockwise, and the copper foil is gone.

Gunther Hartung
Inverness, FL

NEW PRODUCTS

Continued from page 23

- SPI, I²C, and parallel interfaces
- -40°C ~ +85°C operating temperature
- Low power: 2.4~3.6 VDD supply voltage
- RoHS compliant

For more information, contact:
Newhaven Display Int.
http://bit.ly/SlimOLEDsNuts-Volts

READER FEEDBACK
Continued from page 7

are useful and pretty. I have a lovely 1940 wooden radio, but I put Bluetooth inside.

I also made a wooden clock using recycled 1960's counter boards with round Nixie tubes, but it now gets its timing pulse from a microcontroller.

If you'd like, you can check out my retirement hobby projects at **projects.worsleyassociates.com.**

John Saunders
San Diego, CA

Thanks, John. I like the idea of recycling old electronics!

Bryan Bergeron
Editor

■ LOOK FOR   ■ SEARCH FOR   ■ FIND
Find your favorite advertisers here!
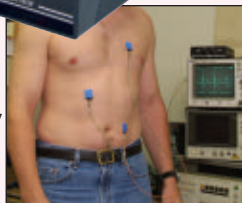
ADvertiser INDEX

Valentine's Day Goodies!

Electrocardiogram ECG Heart Monitor

✔ **Visible and audible display of your heart rhythm!**
✔ **Bright LED "Beat" indicator for easy viewing!**
✔ **Re-usable hospital grade sensors included!**
✔ **Monitor output for professional scope display**
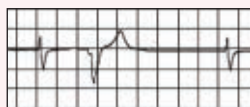✔ **Simple and safe 9V battery operation**

February is the month for Valentine's Day, and what a great time to think of your heart! Not how many times it's been broken, not how many times it's fallen head over heels in love, but how it actually works... and how it's doing these days! Not only will building an actual ECG be a thrill, but you'll get hands-on knowledge of the relationship between electrical activity and the human body! Each time the human heart beats, the heart muscle causes small electrical changes across your skin. By monitoring and amplifying these changes, the ECG1C detects the heartbeat and allows you to accurately display it, and hear it, giving you a window into the inner workings of the human heart and body!

Use the ECG1C to astound your physician with your knowledge of ECG/EKG systems. Enjoy learning about the inner workings of the heart while, at the same time, covering the stage-by-stage electronic circuit theory used in the kit to monitor it. The three probe wire pick-ups allow for easy application and experimentation without the cumbersome harness normally associated with ECG monitors.

One of our engineers/guinea pigs, checking his heart!

Look what I found!

The documentation with the ECG1C covers everything from the circuit description of the kit to the circuit description of the heart! Multiple "beat" indicators include a bright front panel LED that flashes with the actions of the heart along with an adjustable level audio speaker output that supports both mono and stereo hook-ups. In addition, a monitor output is provided to connect to any standard oscilloscope to view the traditional style ECG/EKG waveforms just like you see in a real ER or on one of the medical TV shows! See the display to the right? That was me, when I noticed some skipped beats in my pulse! An immediate cardiac check found I had Trigeminy, or PVCs that occur at intervals of 2 normal beats to one PVC! And I saw it with our ECG1 kit!

The fully adjustable gain control on the front panel allows the user to custom tune the differential signal picked up by the probes giving you a perfect reading and display every time! 10 hospital grade re-usable probe patches are included together with the matching custom case shown. Additional patches are available in 10-packs. Operates on a standard 9VDC battery (not included) for safe and simple operation. Note, while the ECG1C professionally monitors and displays your heart rhythms and functions, it is intended for hobbyist usage only. If you experience any cardiac symptoms, seek proper medical help immediately!

ECG1C	Electrocardiogram Heart Monitor Kit With Case & Patches	**$44.95**
ECG1WT	Electrocardiogram Heart Monitor, Factory Assembled & Tested	**$89.95**
ECGP10	Electrocardiogram Re-Usable Probe Patches, 10-Pack	**$4.95**

Electronic Love Tester

✔ **10 LED love scale display!**
✔ **Audible love level sound!**
✔ **Great party fun!**
✔ **Heart shaped PC board!**

This uniquely shaped "Love Tester" is the ultimate gag for any couple! Designed to check your love life, each partner holds one end of the tester PCB at the appropriate male and female touch pads. Then they romantically join hands and watch the results on the love meter! 10 green, yellow, and red LEDs act like a scale, and just like the carnival, when it hits the top they flash, indicating you're a red hot couple! There is also an audible alarm that changes with the "love level". Next time the party isn't going anywhere, bring this out, it's a riot! Compatible with all couples! Runs on a standard 9V battery (not included)

MK149	Electronic Love Tester Kit	**$16.95**

LED Magic Wand

✔ **Message floats in air!**
✔ **Fun at concerts & events!**
✔ **High visibility red LEDs**
✔ **Pre-programmed or custom messages**

Use the "Magic Wand" to display your true feelings! Simply shake it back and forth and brilliant messages seem to appear in mid-air! Six high intensity LEDs are microprocessor controlled to display messages and graphics that are pre-programmed into the wand. You can also custom program a message of your choice! From amazing your friends, making a statement at a concert, or simply telling your loved one how you feel, the message wand can't be beat! Runs on two AAA batteries (not included), and features auto power-off.

MK155	LED Magic Wand Kit	**$17.95**

LED Flashing Heart

✔ **28 brilliant red LEDs!**
✔ **Unique dual heart design!**
✔ **Freestanding mount!**
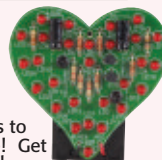
What a way to display your feelings to that very special person in your life! Get out your soldering iron and dazzle her with this unique dual heart electronic display that you can say you built yourself!

28 brilliant red LEDs are formed into two separate heart designs on the heart shaped PC board creating a great flashing display that's easy to assemble. Built-in battery holder base makes the sweetheart freestanding and perfect for a desk or table. Runs on a standard 9V battery (not included). Measures 2.4" x 2.4" x 1.2".

MK101	LED Flashing Sweetheart Kit	**$6.95**

SMT LED Heart Display

✔ **Alternating flashing!**
✔ **6 super bright SMT LEDs!**
✔ **Learn all about SMT!**
✔ **Definitely gets her attention!**
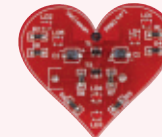
This cute little kit gives you a distinctive red display using 6 Surface Mount (SMT) LEDs. The PC board is in the shape of a red heart. The small size makes it perfect to be used as a badge or hanging pendant around your neck. Even better as an illuminated attention-getting heart to accompany a Valentines Day card! Makes a great SMT learning kit to bring you into the world of SMT technology, design, and hands-on soldering and troubleshooting. Don't worry, extra SMT parts are included just in case you lose or damage any! Runs on a small CR2025/32button cell (not included). Measures 1.9" x 1.7" x .3".

MK144	SMT Flashing Heart Kit	**$11.95**

Ultimate 555 Timers

This new series builds on the classic UT5 kit, but takes it to a whole new level! You can configure it on the fly with easy-to-use jumper settings, drive relays, and directly interface all timer functions with onboard controls or external signals.

UT5A
UT5AS
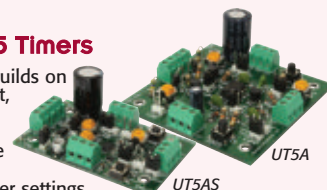
All connections are easily made though terminal blocks. Plus, we've replaced the ceramic capacitor of other timer kits with a Mylar capacitor which keeps your timings stable over a much wider range of voltages! Available in through hole or surface mount versions! Visit www.ramseykits.com for version details.

UT5A	Through Hole 555 Timer/Osc Kit	**$21.95**
UT5AS	SMT 555 Timer/Osc Kit	**$29.95**

Touch Switch

Touch on, touch off, or momentary touch hold, it's your choice with this little kit! Uses CMOS technology. Actually includes TWO totally separate touch circuits on the board! Drives any low voltage load up to 100mA. Runs on 6-12 VDC.

TS1	Touch Switch Kit	**$9.95**

Doppler Direction Finder

Track down jammers and hidden transmitters with ease! 22.5 degree bearing indicator with adjustable damping, phase inversion, scan and more. Includes 5 piece antenna kit. Runs on 12VDC vehicle or battery power.

DDF1	Doppler Direction Finder Kit	**$169.95**

Digital Recorder/Player

This little board records and plays back up to 8 minutes of audio! Features a on-board electret mic as well as line input and output and speaker output! Adjustable sample rate to match your application. Runs on 9-12VDC.

K8094	Audio Storage Module Kit	**$32.95**

HV Plasma Generator

Generate 2" sparks to a handheld screwdriver! Light fluorescent tubes without wires! This plasma generator creates up to 25kV at 20kHz from a solid state circuit! Build plasma bulbs from regular bulbs and more! Runs on 16VAC or 5-24VDC.
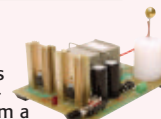
PG13	HV Plasma Generator Kit	**$64.95**

Speedy Speed Radar Gun

Our famous *Speedy* radar gun teaches you doppler effect the fun way! Digital readout displays in MPH, KPH, or FPS. You supply two coffee cans! Runs on 12VDC or our AC121 supply.

SG7	Speed Radar Gun Kit	**$74.95**

Tickle-Stick Shocker

The kit has a pulsing 80 volt tickle output and a mischievous blinking LED. And who can resist a blinking light and an unlabeled switch! Great fun for your desk, "Hey, I told you not to touch!" Runs on 3-6 VDC.
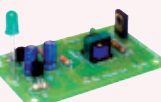
TS4	Tickle Stick Shocker Kit	**$9.95**

Precision PC Plane Antennas

Our LPY series PC antennas continue to be the favorite for virtually all RF and wireless applications. From microwave links, wireless mics, to RFID, we've got you covered. Check our site for details!

LPYSeries	Precision PC Plane Antennas from **$29.95**	

Stereo Audio Gain Controller

✔ **Stereo audio processing while preserving audio dynamics!**
✔ **True stereo control keeps virtual sonic source location intact!**
✔ **Auto-bypass restores original levels when power is turned off!**
✔ **Built-in bar graph indication of signal level with display mute!**

Your Audio Engineer!

The SGC1 is one of our latest kits, and provides a great solution to the age-old problem: how can we easily correct inconsistent audio levels without negatively affecting the dynamics of the audio signal? The SGC1 circuit implements a principle known as the "Platform Gain Principle," which was originally developed by CBS Labs (what we now know as CBS in the TV and radio world) to allow transmitted audio levels to be automatically adjusted to keep them within a desired range.

Think of it like an audio engineer, constantly adjusting the output level in order to limit highs that would be too loud while boosting lower levels so that they can still be heard. You may think "oh, this is just another limiter/compressor!" Not so! Here's the real trick: keeping the full dynamic range ratio of the output signal the same as the original input - something the typical limiter/compressor can only dream of doing! The SGC1 can be placed in just about any standard analog stereo line level audio circuit (the red and white RCA connectors or the mini-phone connector) to keep the audio level within the desired range. It's also the perfect addition to any of our hobby kit transmitters, allowing you to match levels between different audio sources while keeping lows audible and preventing the highs from overdriving. The SGC1 makes a great addition to any audio system where you need to keep levels from different sources under control, but still make sure they all sound great! In addition to its useful basic function and great audio performance, the SGC1 also boasts a front panel LED meter to give an indication of the relative level of the input signal, plus a level control (also on the front panel) that allows you to adjust the controller to the min/max center point of your desired level range. And yes, it is a *Stereo Gain Controller!* Meaning that the levels of both the left and right channels are monitored and adjusted equally, thereby maintaining the relative virtual position of things like instruments, singers and speakers! The entire unit is housed in a slim attractive black textured aluminum case that is sure to complement your studio or home theatre. If you're looking for perfect audio levels, hire a broadcast audio engineer, but if that doesn't fit your budget, the SGC1 is the next best thing! Includes 15VDC world-wide power adapter.

SGC1 Stereo Audio Platform Gain Controller Kit **$179.95**

8-Channel Remote Ethernet Controller

Now you can easily control and monitor up to 8 separate circuits via the standard Ethernet network in your home or office. Connection wise it couldn't be simpler. The controller functions as an IP based web server, so it can be controlled by any internet browser that can reach your network! There are no drivers or proprietary software required, just access the controller like any web page from your PC, laptop, or even your smartphone!

Security is assured allowing up to 4 separate user credentials. The controller can be set to a specific static IP within your network subnet or can be set to DHCP (auto negotiate). The controller can even be programmed to send you an email to notify and confirm power up and status changes!

To simplify the connection of your equipment to the controller, 8 separate and isolated relay outputs are provided! This gives you internet or network control of up to 8 separate functions. No need to worry about interfacing a logic high or logic low, or burning up the interface! The applications are endless! From something as simple as turning on and monitoring lights at your house with a normal latched closure to advanced control of your electronic gadgets, radio equipment, or even your garage door!

Each relay contact is rated at 12A at 30VDC or 16A at 230VAC. Each of the 8 channels has built-in timer and scheduler programs for day, weekend, working days, every day, and every day except Sunday. Relay control functions are programmable for on, off, or pulse (1-99 seconds, 1-99 minutes, or 1-99 hours). In addition to control functions, the web interface also displays and confirms the status of each channel. Each channel can be custom labeled for your specific function name. The controller operates on 12VDC or 12VAC at 500mA or our new AC121 global 12VDC switching power supply to the right above. Factory assembled, tested, and ready to go! Even includes a Cat-5 cable!

VM201 8-Channel Remote Ethernet Controller, Factory Assembled & Tested **$169.95**

Air Blasting Ion Generator

Generates negative ions along with a hefty blast of fresh air, all without any noise! The steady state DC voltage generates 7.5kV DC negative at 400uA, and that's LOTS of ions! Includes 7 wind tubes for max air! Runs on 12-15VDC.

IG7 Ion Generator Kit **$64.95**

Broadband RF Preamp

Need to "perk-up" your counter or other equipment to read weak signals? This preamp has low noise and yet provides 25dB gain from 1MHz to well over 1GHz! Output can reach 100mW! Runs on 12 volts AC or DC or the included 110VAC PS. Assmb.

PR2 Broadband RF Preamp **$69.95**

Tri-Field Meter Kit

THE GHOST DETECTOR!

"See" electrical, magnetic, and RF fields as a graphical LED display on the front panel! Use it to detect these fields in your house, find RF sources, you name it. Featured on CBS's Ghost Whisperer to detect the presence of ghosts! Req's 4 AAA batteries.

TFM3C Tri-Field Meter Kit **$74.95**

Passive Aircraft Receiver

The hit of the decade! Our patented receiver hears the entire aircraft band without any tuning! Passive design has no LO, therefore can be used on board aircraft! Perfect for airshows, hears the active traffic as it happens! Available kit or factory assembled.

ABM1 Passive Aircraft Receiver Kit **$89.95**

Electret Condenser Mic

This extremely sensitive 3/8" mic has a built-in FET preamplifier! It's a great replacement mic, or a perfect answer to add a mic to your project. Powered by 3-15VDC, and we even include coupling cap and a current limiting resistor! Extremely popular!

MC1 Mini Electret Condenser Mic Kit **$3.95**

Laser Trip Sensor Alarm

True laser protects over 500 yards! At last within the reach of the hobbyist, this neat kit uses a standard laser pointer (included) to provide both audible and visual alert of a broken path. 5A relay makes it simple to interface! Breakaway board to separate sections.

LTS1 Laser Trip Sensor Alarm Kit **$29.95**

12VDC Regulated Switching Supply

Go green with our new 12VDC 1A regulated supply. Worldwide input 100-240VAC with a Level-V efficiency! It gets even better, includes DUAL ferrite cores for RF and EMI suppression. All this at a 10 buck old wallwart price! What a deal!

AC121 **12VDC 1A Regulated Supply** **$9.95**

12VDC Worldwide Supply

It gets even better than our AC121 above! Now, take the regulated Level-V green supply, bump the current up to 1.25A, and include multiple blades for global country compatibility! Dual ferrite cores!

**PS29** **12VDC 1.25A Global Power Supply** **$19.95**

Sniff-It RF Detector Probe

Measure RF with your standard DMM or VOM! This extremely sensitive RF detector probe connects to any voltmeter and allows you to measure RF from 100kHz to over 1GHz! So sensitive it can be used as a RF field strength meter!

RF1 **Sniff-It RF Detector Probe Kit** **$27.95**

The Learning Center!

PL130A
PL200
PL300
SM200K
AMFM108K
SP1A
PL500
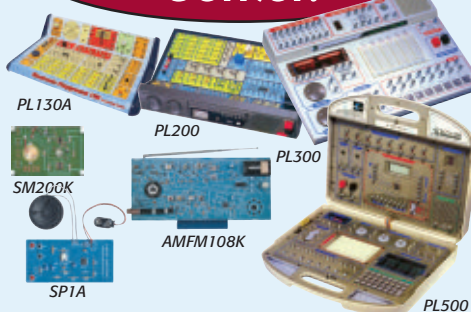
Fun Electronic Learning Labs

✔ **Learn and build!**
✔ **130, 200, 300, & 500 in one labs!**
✔ **Practical through hole and SMT soldering labs!**
✔ **Integrated circuit AM/FM radio lab!**
✔ **Super comprehensive training manuals!**

Starting out our "All in One" series, the PL130A, gives you 130 different electronic projects, together with a comprehensive 162 page learning manual. A great start for the kids...young and old! Next, check out the PL200, that gives you 200 very creative and fun projects, and includes a neat interactive front panel with 2 controls, speaker, LED display and a meter. From there, step up to our PL300, which gives you 300 separate electronic projects along with s 165 page learning and theory manual. The PL300 walks you through the learning phase of digital electronics. If you're looking for the ultimate lab kit, check out our PL500. It includes a whopping 500 separate projects, a 152 page starter course manual, a 78 page advanced course manual, and a 140 page programming course manual! The PL500 covers everything from the basics to digital programming!

If you are looking to either learn or hone up on your through hole or SMT soldering skills check our SP1A and SM200K Practical Soldering Labs. You will be a soldering master in no time!

We make it easy to learn IC's while at the same time, building a neat AM/FM radio with our AMFM108K AM/FM IC lab kit. You will have a blast AND learn!

PL130A	130-In-One Lab Kit	**$39.95**
PL200	200-In-One Lab Kit	**$84.95**
PL300	300-In-One Lab Kit	**$109.95**
PL500	500-In-One Lab Kit	**$249.95**
SP1A	Through Hold Soldering Lab	**$9.95**
SM200K	SMT Practical Soldering Lab	**$22.95**
AMFM108K	AM/FM IC Lab Kit & Course	**$36.95**

800-446-2295
www.ramseykits.com

Prices, availability, and specifications are subject to change. According to Robin we are not responsible for typos, stupids, printer's bleed, or broken hearts. Possession of this Valentine's Day ad will not guarantee you a date! Visit www.ramseykits.com for the latest pricing, specials, terms and conditions. Copyright 2015 Ramsey Electronics®.. so there!

RAMSEY ELECTRONICS®
590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560

WHAT'S HOT

Test & Measurement and Other Essentials!

OSCILLOSCOPES

Rigol

RIGOL's test & measurement instruments compete with industry leaders – but more affordably: Analog & Mixed-signal Scopes to 1GHz with AWG, Function/AWGs from 20MHz-350MHz RF Signal Gens to 6GHz, Pwr Supplies, DMMs, & Spectrum Analyzers.

Owon

OWON's affordable, reliable, easy-to-use precision benchtop & handheld scopes are unbeatable in their price range. Battery powered and portable options for field use. Owon's Triple Output Power Supplies offer remote control & preset configurations.

Pico

PICO TECHNOLOGY - the world's best PC-based oscilloscopes & data acquisition equipment with the performance of good bench scopes; compact & lightweight - ideal for field work. Award-winning automotive scopes find engine & electric faults quickly.

MISC.

USB Control

Need to interface a PC USB port to other equipment? We have modules for USB-I2C with additional I/O, USB data acquisition, USB-serial (some within the connector itself!), the best and most reliable USB-serial cables, and USB isolators to eliminate ground loops.

Power Supplies

We stock numerous power supplies from simple "wall-warts" for low-voltage/low-current needs, to more complex bench-top supplies with single or multiple outputs, programmable supplies, flexible output current/voltage supplies with multiple communication options.

3D Printers

3D printing makes product and device development faster and cheaper – rapidly build concept parts and projects. We have the latest 3D printers from industry leaders MakerBot and Afinia, as well as the best 3D printing filament available for creating high quality parts quickly!

CIRCUIT/PCB TEST

ABI Electronics

ABI ELECTRONICS - PCB test/repair: BoardMaster 8000: easy-use PCB test system; JTAGMaster: programs/tests via JTAG port; RevEng Schematic Learning System: generates schematics from PCBs; SENTRY: detects counterfeit components.

MQP

MQP ELECTRONICS – USB2.0 Analyzers /Gens, Protocol/Electrical Test equipment, offering unbeatable value/performance ratio and Vbus monitoring. GraphicUSB software offers full analysis of std USB2.0 protocol with Class Analysis options.

Cosview

COSVIEW - range of extremely affordable USB-connected inspection microscopes & stands, very useful for examining printed circuit boards and small integrated circuits for production faults or part number markings. Polarizing filter option which excludes surface glare is available.